UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Study of fluid dynamics during the depressurization step in the DELOS-SUSP process

David Piña Muñoz

Supervisors

Alba Córdoba Insensé
Robert Castilla Lopez

Tutor

Alba Àgueda Costafreda

Universitat Politècnica de Catalunya, Oct. 2019
Escola d'Enginyeria de Barcelona Est (EEBE)

Master's Thesis in Chemical Engineering

MASTER'S THESIS IN CHEMICAL ENGINEERING

12 ECTS

Study of fluid dynamics during the depressurization step in the DELOS-SUSP process

DAVID PIÑA MUÑOZ

Nanomol Technologies S.L.
Departament de mecànica de fluids, UPC

UNIVERSITAT POLITÈCNICA DE CATALUNYA
ESCOLA D'ENGINYERIA DE BARCELONA EST (EEBE)

Barcelona, Spain 2019

Abstract

DELOS-SUSP is a one-step, compressed fluids-based process for the production of vesicles. In the last years, this process has been used for the generation of *Quatsomes*, which are thermodinamically stable vesicles used as drug delivery systems (DDs). Their physicochemical properties are greatly influenced by the parameters used in their production process. In particular, recent studies indicate that, the size and polydispersity of these vesicles, are affected by the depressurization flow rate, Q. These physicochemical properties are directly linked with their viability as efficient DDs. Consequently, a proper study of the relationship between Q and the *Quatsomes* properties is required for the optimization of the process.

In this master's thesis, the mixing of the fluids during the depressurization step of DELOS-SUSP at different Q has been studied by OpenFOAM, a Computational Fluid Dyanamics (CFD) software. In this context, the spatial and temporal distribution of the ethanol concentrarion (one of the components of the process) has been analyzed. In addition, this work provides for the first time, a detailed protocol of the incorporation of DELOS-SUSP in a CFD environment, which can be useful for further studies in the field.

Resum

El DELOS-SUSP és un procés d'una sola etapa per a la producció de vesícules. La metodologia d'operació es basa en l'ús de fluids comprimits. En els últims anys, aquest procés s'ha fet servir per a la generació de *Quatsomes*, vesícules termodinàmicament estables i que s'utilitzen com a transportadors de fàrmacs (DDs). Les propietats fisicoquímiques dels *Quatsomes* estan estretament lligades amb el procés de producció dels mateixos. Concretament, estudis recents han determinat que la mida i la polidispersitat, propietats lligades amb la seva eficiència com a DDs, es poden relacionar amb el cabal de despressurització del procés, Q. Conseqüentment, a fi d'optimitzar el procés, es requereix l'estudi de la relació de Q amb les propietats dels *Quatsomes*. En aquesta tesi de màster s'ha estudiat, mitjançant OpenFOAM, un software de fluidodinàmica computacional (CFD), el mesclat dels diferents fluids durant l'etapa de despressurització del DELOS-SUSP. Sota aquest context, s'ha analitzat la distribució temporal i espacial de la concentració d'etanol (un dels components del procés). A més, aquest treball proporciona, per primer cop, un protocol detallat sobre la incorporació del DELOS-SUSP en l'entorn del CFD, que podrà ser útil en propers estudis en el camp.

Resumen

El DELOS-SUSP es un proceso de una sola etapa para la producción de vesículas. La metodología se basa en el uso de fluidos comprimidos. En los últimos años, este proceso se ha utilizado para la generación de *Quatsomes*, vesículas termodinámicamente estables y que se utilizan como transportadores de fármacos (DDs). Las propiedades fisicoquímicas de los *Quatsomes* están estrechamente ligadas con el proceso de producción de los mismos. Concretamente, estudios recientes han determinado que el tamaño y polidispersidad, propiedades relacionadas con su eficiencia como DDs, se pueden relacionar con el caudal de despresurización Q. Consecuentemente, con la finalidad de optimizar el proceso, se requiere el estudio de la relación de Q con las propiedades de los *Quatsomes*. En esta tesis de máster se ha estudiado, mediante OpenFOAM, un software de fluidodinámica computacional (CFD), el mezclado de los diferentes fluidos durante la etapa de despresurización del DELOS-SUSP. Bajo este contexto, se ha analizado la distribución espacial y temporal de la concentración de etanol (uno de los componentes del proceso). Además, este trabajo proporciona, por primera vez, un protocolo detallado sobre la incorporación del DELOS-SUSP en un entrono CFD, que podrá ser útil en próximos estudios en el campo.

Acknowledgements

A la pregunta: *¿Cómo un graduado en Nanociencia está haciendo y máster en Ingeniería Química y un TFM sobre mecánica de fluidos computacional?*, la respuesta es : durmiendo poco y con mucha ayuda. Este proyecto ha sido difícil para mi desde el principio. Sin haber tocado Linux en mi vida, (y OpenFOAM ya, ni digamos) me lancé a una aventura arriesgada. Aunque al final, lo importante es lo que he aprendido, que es muchísimo, un poquito de cada uno de vosotros.

Antes que a nadie, muchas gracias a la Dr. Alba Córdoba y al Dr. Robert Castilla por supervisarme y hacer que esto saliera adelante. No hubiese sido posible sin vosotros. Gracias también al Dr. Gustavo Raush, porque su ayuda semanal también ha sido de valor incalculable. También gracias a mi tutora, la Dr. Alba Àgueda, por sus palabras amables y útiles aportaciones.

Gracias al ICMAB, al grupo Nanomol, y a la ICTS/NANBIOSIS, concretamente a la unidad de *Biomaterial Processing and Nanostructuring Unit (U6)*, perteneciente al CIBER-BBN y localizada en el ICMAB-CSIC, por permitirme desarrollar la parte experimental. Gracias al servicio de microscopia de la UAB por las imágenes de criomicroscopía. También a Nanomol Technologies, especialmente a Josep por los datos para el experimental! Gracias también a la Dr. Nora Ventosa porque este proyecto es fruto de sus ideas.

También dar las gracias al Departament de Mecànica de Fluids de la UPC-ESEIAAT y al centro LABSON por prestarme sus *clusters* para correr las simulaciones.

Escribir el TFM también ha sido un poco drama. Muchísimas gracias Adri por los muchísimos ratos perdidos conmigo para ayudarme con LaTeX. Este trabajo sería indudablemente más feo sin tu increíble paciencia. Tu sabes lo mucho que te lo agradezco Dri. Y cómo olvidar a mi compi de despacho, Angel, gracias por tu ayuda con el endiablado Blender y a Guillem, por ese super ojo de robot buscando puntos y espacios! Y gracias a mi Paulisima, por dar serenidad cuando era necesario y por las muy necesarias correcciones.

Finalmente, gracias a mis supernanoninos. De verdad, los días más estresantes se hacen mucho más fáciles con vosotros al lado. Estar en un ambiente así te da la vida. Sabéis que os quiero a cada uno de vosotros. Y, gracias a David, por aguantar las pataletas y las malas caras, siempre con buenas palabras. Tengo mucha suerte.

List of Figures

List of Tables

Abbreviations and symbols

P pressure. 3, 6, 20

Q depressurization flow rate. 1, 7, 12, 22, 24–29, 34, 36–38, 40

T temperature. 3, 6

U velocity. 9, 20

$\boldsymbol{X_{CO2}}$ molar fraction of CO_2. 5, 24, 43

3D three dimensional. 15, 17

APIs active pharmaceutical ingredients. 3

BCs boundary conditions. XIV, 8, 9, 16, 20, 21

CFD computational fluid dynamics. I, III, 7, 8, 12, 28, 47

CFs Compressed fluids. 3, 4

$\boldsymbol{CO_2}$ carbon dioxide. I, XII, 4, 5, 7, 13, 22–25, 34–37, 41–43, 47

cryo-TEM cryogenic Transmission Electron Microscopy. 2, 26, 27, 45

CTAB Cetyl Trimethyl Ammonium Bromide. 2, 5, 41, 43

D diffusion coefficient. 45

DDs drug delivery systems. 1, 6, 28

DELOS Depressurization of an Expanded Organic Solution. 4

DELOS-SUSP Depressurization of an Expanded Liquid Organic Solution over a Suspension. 4–7, 12, 15, 18, 22, 24, 28, 40–42, 47

DLS Dynamic Light Scattering. 2, 26, 45

EtOH ethanol. 22, 23, 43

GUI Graphical User Interface. 8

N_2 molecular nitrogen. 5, 42, 43

OpenFOAM Open-source Field Operation And Manipulation. I–III, V, VII, X, XVI, 8–15, 20–23, 41, 47, 48

P_W working pressure. 4, 5, 24, 43

Pc critical pressure. 3

PdI polydispersity index. v, 26, 46

SUVs small unilamellar vesicles. 5

T_W working temperature. 4, 5, 24, 43

Tc critical temperature. 3

VOF volume of fluid. 14

Contents

1 Introduction and theoretical background

For many years now, it is believed that nanotechnology has the potential to drastically change the biomedicine field. As a matter of fact, diagnose and treatment of some illnesses have already been improved by the use of nanomaterials, which are synthetic, nanoscopic materials. These materials are able to interact with its surroundings in a way that could not be achieved by its macroscopic counterparts [1].

In this framework, there is a huge activity in the research of new drug delivery systems (DDs). DDs are synthetic nanostructured materials that can be used for transporting, in a safer and more efficient way, a therapeutic or diagnostic agent to the desired organ or tissue [2, 3]. Their most relevant advantages can be summarized as:

- Protection of the pharmaceutical agent from body biodegradation.
- Increase of the assimilation and absorption of the pharmaceutical through the different body barriers.
- Improvement of cellular penetration and biodistribution of the biomolecule.
- Reduction of the side effects. Doses can be minimized since the conduction of the active towards a selective target increases its efficiency and avoids its bioacumulation in non-targeted tissues.

Among the different types of DDs, vesicles are nanometric or micrometric spheres formed by amphiphilic molecules, that are self-assembled in bilayered structures. These hollow spheres, normally prepared in water or in an aqueous media, enclose an aqueous volume inside, allowing vesicles to entrap water-soluble molecules in its lumen, and apolar ones between their bilayers. [4].

In this work, the formation of non-liposomal vesicles using the *Depressurization of an Expanded Liquid Organic Solution over a Suspension* (DELOS-SUSP) process will be studied. The size and morphology of these vesicles have been seen to be affected by process parameters, such as the depressurization flow rate (Q), which will be studied by means of CFD tools.

1.1 Background

1.1.1 *Quatsomes*: stable, non-liposomal, vesicular systems

Among the different vesicular systems, liposomes, made of phospholipids, have been the most consistently used and studied in nanomedicine since Dr. Alec Bangham *et al.* first reported their discovery in 1964 [5]. Their most important strength relies in their huge biocompatibility. Nevertheless, some of their properties hinders their utility as DDs. Liposomes correspond to metastable, kinetically trapped states, which are formed by the input of external energy on a planar lamellar phase (i.e., sonication, extrusion, etc.) [6]. As a result, their thermodinamically stable state is this planar, lamellar structure in which liposomes precipitate eventually, losing their advantages as DDs.

With the purpose of proposing a viable, stable-in-time alternative to the liposomes, *Nanomol* group obtained non-liposomal nanoscopic vesicles using cholesterol and Cetyl Trimethyl Ammonium Bromide (CTAB), called ***Quatsomes*** [7]. Thus, by definition, a *Quatsome* is any vesicular system made of -sterol molecules, traditionally cholesterol, and quaternary ammonium surfactants, traditionally CTAB (Figure 1.1). Dynamic Light Scattering (DLS) and cryogenic Transmission Electron Microscopy (cryo-TEM) measurements (Figure 1.2) revealed that *Quatsomes* prepared by the DELOS-SUSP process, explained in the following sections, have an outstanding vesicle to vesicle homogeneity regarding size and lamellarity [7].

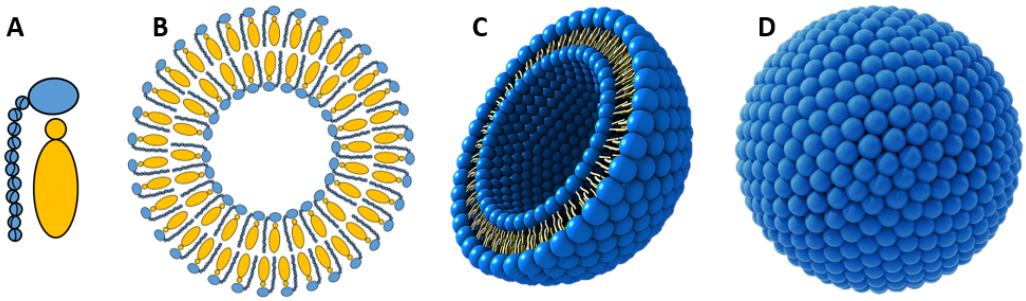

Figure. 1.1: Schematic representation of a Quatsome and its building blocks. From left to right: A: structural block of CTAB (blue) and cholesterol (yellow); B: cross-section representation of the building blocks forming the vesicle; C,D: representation of the Quatsome semisphere and sphere. [7]

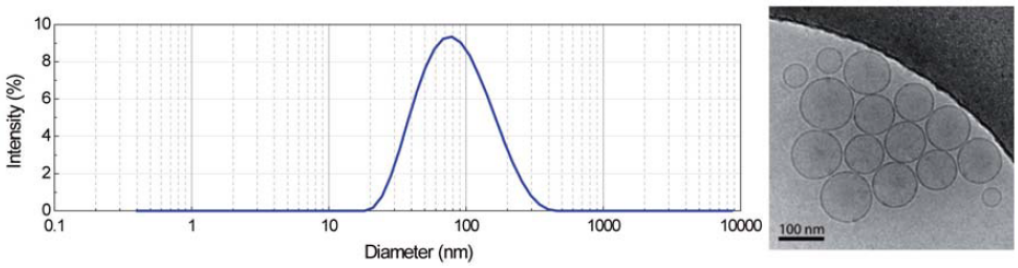

Figure. 1.2: Particle size distribution (left) and cryo-TEM image (right) of Cholesterol/CTAB *Quatsomes* nanovesicles in water [8]

Furthermore, *Quatsomes* are thermodinamically stable and can be stored for years at room temperature. Thus, their unique properties of stability, size and homogeneity, coupled with their capability of entrapping active pharmaceutical ingredients (APIs) or diagnostic agents makes them promising systems to be used in biopharmaceutical applications [9, 10].

1.1.2 Advantages of CFs in material processing

Compressed fluids (CFs)-based methodologies have been gaining ground during the last 30 years as promising alternatives to conventional methodologies for the preparation of nanostructured materials [11, 12]. CFs are defined as substances that at normal conditions of pressure (P) and temperature (T) exist as gases but with increased P can be converted into liquids or supercritical fluids. The supercritical region is achieved when the substance is exposed to conditions above its critical pressure (Pc) and critical temperature (Tc). The most important feature within the supercritical region is that there is no phase boundary between the gas and liquid phases. The consequence is that supercritical fluids have properties which are "hybrids" of those normally associated with liquids and gases and which are continuously adjustable from gas to liquid with small pressure and temperature variations. Thus, the viscosities and diffusivities are similar to those of the gas phase while the density is closer to that of the liquid.

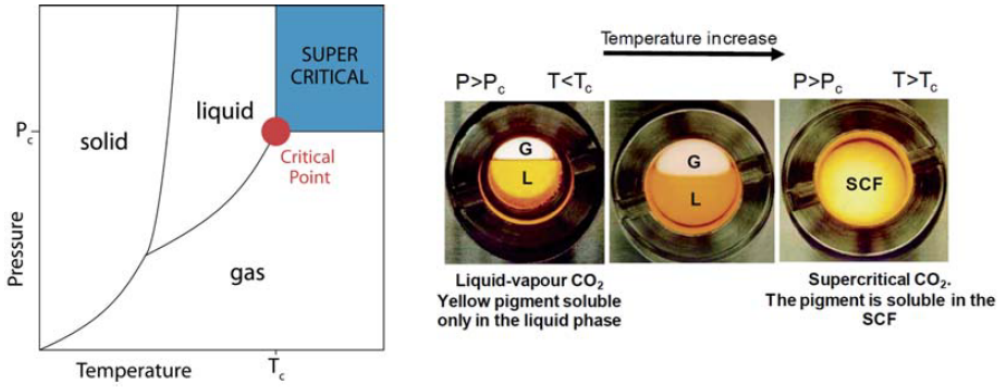

Figure. 1.3: Phase diagram of a compressed fluid (CO_2) (left) and solvent power of CO_2 as a function of its state(right)

1.1.3 DELOS-SUSP process: Preparation of vesicles using CFs

In the context of the thriving CFs-based technologies, *Nanomol* group developed a new procedure based on the use of compressed carbon dioxide (CO_2) called Depressurization of an Expanded Organic Solution (DELOS) for the production of micron and sub-micron sized particles with a high control of their physicochemical characteristics, such as narrow particle size distribution or high polymorphic homogeneity [13]. The DELOS uses CO_2 as a co-solvent, being miscible in the organic solution containing the solute to be micronized [13, 14]. This process uses milder conditions ($<$ 10 MPa, $<$ 308 K) than others, more standardized methodologies based on CFs [15], allowing the processing of heat-labile compounds and reducing the cost of a high pressure plant.

With the experience of the DELOS process, it was later developed the Depressurization of an Expanded Liquid Organic Solution over a Suspension (DELOS-SUSP), a modification of the former, in order to prepare colloidal suspensions. This methodology enabled one-step preparations of cholesterol-rich vesicles [7, 16]. Both methods are patent protected and currently owned by the company Nanomol Technologies S.L. [17, 18].

Briefly, the DELOS-SUSP method (Figure 1.4) consists in loading a solution of the membrane lipidic components (i.e. cholesterol) in an organic solvent into a high-pressure autoclave, previously heated to the working temperature (T_W) (Figure 1.4a). In a second stage, the vessel or autoclave is pressurized with a specific amount of compressed CO_2 until the working pressure (P_W) has been reached (10 MPa) (Figure 1.4b). Finally, in the third

stage, the vesicles are formed by depressurizing the resulting CO_2-expanded solution over an aqueous phase at atmospheric pressure, which contain water soluble surfactants (i.e. CTAB) (Figure 1.4c).

In the depressurization step of the process (Figure 1.4c), a flow of molecular nitrogen (N_2) at P_W is used as a plunger to push down the CO_2-expanded solution and keep the pressure constant inside the pressurized vessel. During the depressurization step, the expanded organic solution experiences a large, abrupt and extremely homogeneous temperature decrease produced by the CO_2 evaporation from the expanded solution. This fact may be the reason of obtaining homogeneous vesicles regarding size, lamellarity and morphology, although the vesicle formation mechanism is currently unknown. No further energy is required for achieving the desired small unilamellar vesicles (SUVs).

A detailed explanation of the formation of vesicular systems by DELOS-SUSP process will be described later on this work, in Section 5.3.

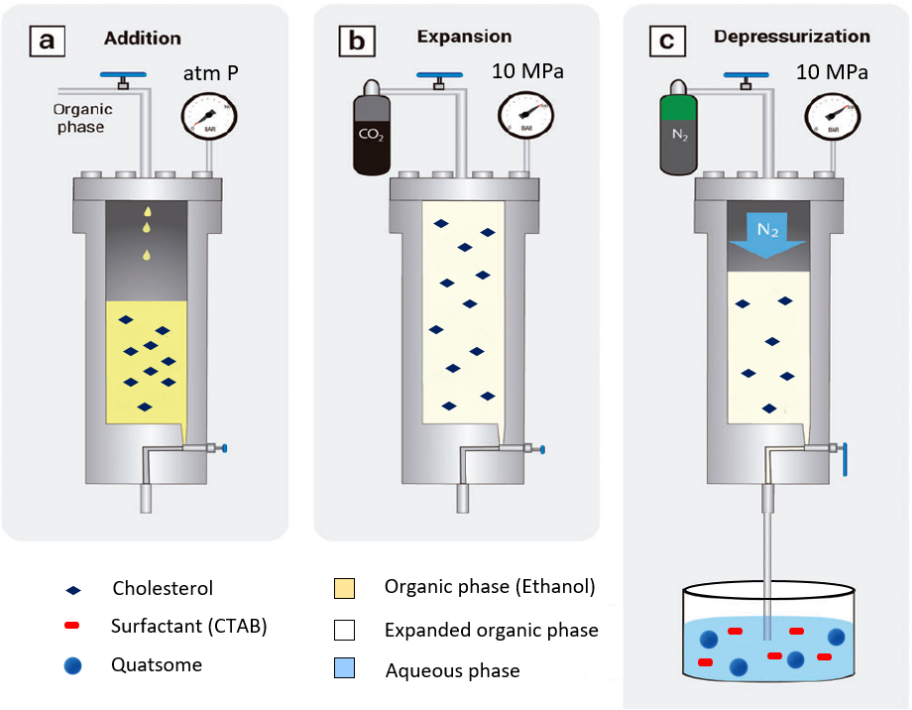

Figure. 1.4: Schematic representation of the DELOS-SUSP process for one-step preparation of vesicles. The procedure includes: (a) the loading of an organic solution of the lipidic membrane components into an autoclave at a T_W and atmospheric pressure; (b) addition of CO_2 to produce a expanded solution with a given molar fraction of CO_2 (X_{CO2}), at P_W and T_W, where the lipidic membrane components remaine dissolved; and finally (c), the depressurization of the expanded solution over an aqueous solution containing the surfactants needed for the self-assembly of the vesicles. Figure adapted from [19].

1.1.4 Impact of the preparation method in *Quatsomes* formation

Quatsomes are physically assembled nanoparticles as opposed to other DDs such as dendrimers or polymeric nanoparticles, where the forming blocks or subunits are chemically assembled through covalent bonding. *Quatsomes*, and in general, most vesicular systems, are formed through self-assembly. Self-assembly is the process in which a disordered system of pre-existing components forms an organized structure as a consequence of specific, local interactions among the components themselves, without external direction. Self-assembled structures are built as the new "conformation" approaches equilibrium, and thus, reducing the free energy of the system [20].

As stated in the previous sections, *Quatsomes* are thermodinamically stable, which implies that for specific conditions of P, T, and composition, the system will eventually tend to the stable *Quatsome* conformation (monodisperse unilamellar spheres, with a diameter of 60 - 70 nm). This modification of the system to a more stable conformation can be achieved because the weak physical interaction between the building blocks allow movement of the molecules, as opposed to what happens in a rigid, chemically bound macromolecule.

However, even if the final properties of the vesicular system are set by thermodynamics, **the initial (first weeks after production) morphological properties of these vesicles are greatly determined by their preparation method**. These initial properties can not be understimated since the main application of *Quatsomes* consists in the encapsulation of pharmaceutical agents, some of which can be fast-degrading biomolecules that needs to be administered fresh. In this context, achieving the desired vesicle properties right after the production is desired. In addition, on a commercial level, it is important that the product reaches its final form as fast as possible in order to avoid extra economical costs in terms of storing and quality control. For this reason, different methods for preparing *Quatsomes* have been explored, the most important ones being the preparation by *Thin Film Hydration*, preparation by sonication and preparation by the already explained DELOS-SUSP. The former ones will not be covered in this work, but detailed information about them can be read in previously published works [8]. Between these three methods, DELOS-SUSP has yielded unprecedented good results in terms of low polydispersity and high homogeneity of the nanoparticles. However, the initial properties of *Quatsomes* can be very different depending on the parameters in their production even when using DELOS-SUSP. For example, increasing or decreasing the temperature of the

system during the depressurization, changing the molar fraction of CO_2 [8, 19] or varying the depressurization flow rate (Q) in the DELOS-SUSP process can produce different *Quatsomes* in terms of quantity and quality. Especially Q has been seen to have a high impact on *Quatsomes* formation. **In particular, previous experimental evidence seems to indicate that vesicles produced at lower Q have lower polydispersities than the ones produced at higher Q.**

1.1.5 Study of nanoparticles formation by computational fluid dynamics

Computational fluid dynamics is defined as a branch of fluid mechanics that uses numerical analysis and data structures to analyze and solve problems that involve fluid flows. Its use is widely spread today. Since the last decade, CPUs have become more powerful and affordable, and because of that, most large companies in industry are taking profit of this tool. Materials science is also onboard the computational fluid dynamics (CFD) trend since it is a powerful ally in the characterization of the processes used to produce subsized structured materials.

In this context, compressed fluids-based technologies for the preparation of nanoparticles have been studied during the last decade. In those researches, experimental results are compared to CFD numerical simulations with the intent to glimpse the formation mechanisms of these particles. Examples of this are the study of the size distibution in TiO_2 nanoparticles by decomposition of an organometallic precursor in supercritical CO_2 [21] or the analysis of the fluid mixing dynamics in the formation of microparticles by Gas Antisolvent Technique [22].

On the other hand, there has also been research in the characterization of the mechanism formation of vesicular systems (mainly liposomes) using CFD analysis. In a work from 2014, the study of the formation and characteristics of liposomes produced in a three-dymensional microfluidic system was supported by the CFD simulations of the water/ethanol profile inside this microfluidic system [23].

Even though this two topics have been studied separately, the literature covering the CFD study of the formation of vesicles using compressed fluid techniques is scarce to non-existent.

1.2 Computational fluid dynamics software: OpenFOAM

Even though the use of CFD is rising in science and business worlds, investing in the required hardware and expensive commercial lincenses is still a hurdle for smaller or mid businesses to use CFD. In that context, open-source softwares provide a cheap approach to simulations compared to commercial softwares. Open-source Field Operation And Manipulation (OpenFOAM) is an open-source CFD software that is, in fact, a *C++* library used to create executables, also called *applications*. There are two categories: *solvers*, that solve specific problems using the governing equations of fluid mechanics, and *utilities*, that perform tasks involving data visualization and manipulation. OpenFOAM has an extensive range of solving features, from complex fluid flows, involving chemical reactions, turbulence and heat transfer, to solid dynamics and electromagnetics [24]. As a counterpart, OpenFOAM, like other open-source softwares, are dependent on a more knowledgeable user than commercial softwares, as more freedom is provided and documentation can be somehow limited or unreliable. Also, the lack of a built-in Graphical User Interface (GUI) can hinder the use of the tool for people with little informatic skills. However, people have been chosing OpenFOAM due to the following reasons:

- Free software. Commercial CFD tools are rather costly.
- Open-source code. It is possible to adapt the code and build new functionalities.
- It is community driven. Different communities are working on its development which enables a fast evolution.
- Can work in parallel with inbuilt *OpenMPI* environment.
- Already used for active flow control implementation.
- Wide range of turbulent models, discretization schemes, linear solvers, boundary conditions (BCs)...
- Compatible with various pre-processed meshing file formats. Also provides output for multiple post-processing softwares.
- Although the software lacks a GUI, it is more or less easy to define parameters due to a rather well organized folder system as shown in Figure 1.5.

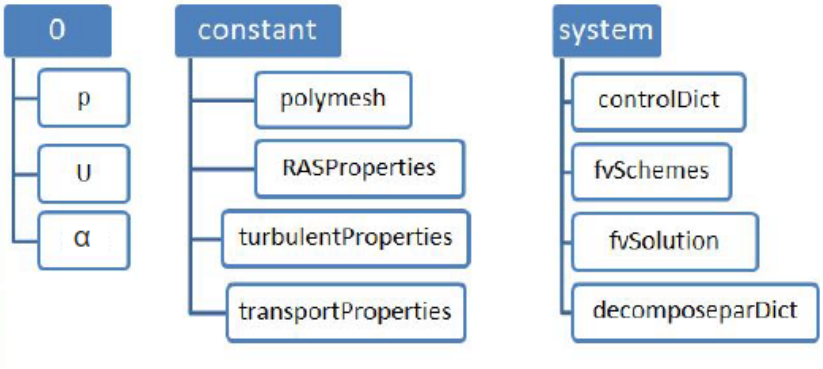

Figure. 1.5: Case structure in OpenFOAM.

The problem that wants to be simulated, named *case* and distributed as shown in the Figure 1.5, can be outlined as follows: the *0* folder contains the files that define the BCs of the parameters and initial values of pressure (P), velocity (U) and, if applicable, composition. Other files can appear in the 0 folder that refer to parameters that are needed for modelling the turbulence of the system. In the case of the present work, we will deal with a triphasic flow but, due to the complexity that entrails solving 3 individual components, no turbulent model will be applied in order to reduce the complexity of the problem and processing times. The *constant* folder contains a mesh of the case geometry as well as the physical properties (kinematic viscosity, density, surface tension...) of the problem fluids. Last, the *system* folder contains the parameters associated with the simulation procedure itself. It contains at least 3 files *controlDict, fvSchemes* and *fvSolution* in which iterations, time stepping, discretization schemes and linear solvers are declared. In the following subsections, a brief summary of each step of the simulation process: (1) pre-processing, (2) solving and (3) post-processing, is briefly described.

1.2.1 Pre-processing

It is the first stage in the development of the *case*. Before the simulation starts, the following aspects must be addressed:

Geometry
The geometry of the problem needs to be designed or drawn. If the geometry is somehow complex (more than a few cubic blocks), a good idea is to create this geometry with another software. OpenFOAM allows the importation of

pre-made geometries (in form of *.stl* surface files) created using third-party software such as *SALOME* or *Blender*.

Mesh generation

A mesh is a defined volume split in several (normally thousands) of individual cells where the simulation occurs. OpenFOAM contains built-in meshing tools such as *blockMesh*, a dictionary used for creating (simple) meshes made of hexes. On the other hand, external geometries created with the third-party software mentioned just above, can afterwards be meshed by using a mesh generator called *SnapphyHexMesh*, which fuses said surfaces with an already created, simpler mesh. An schematic representation of SnappyHexMesh mechanism can be seen in Figure 1.6.

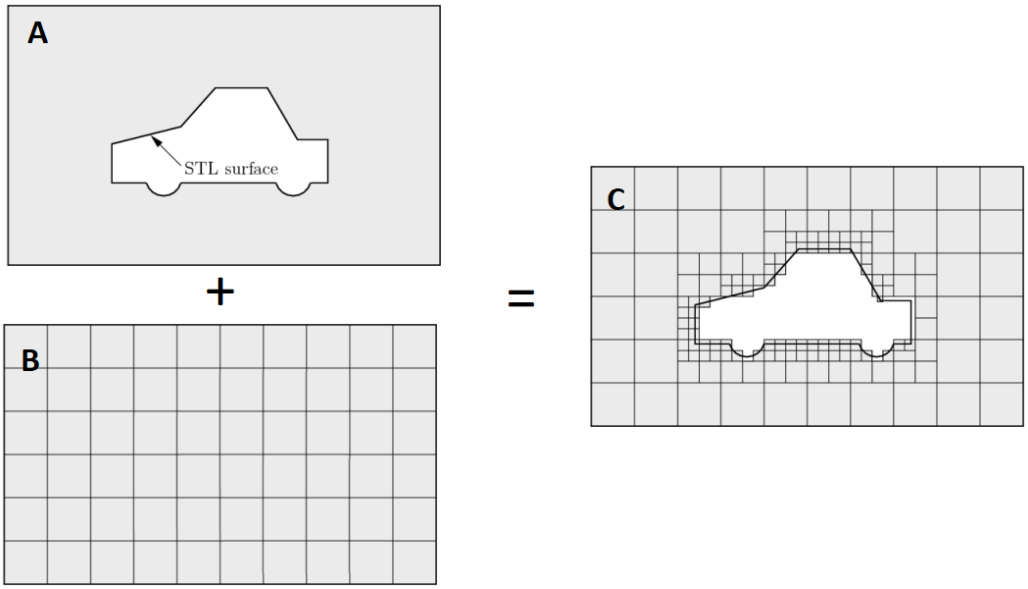

Figure. 1.6: Schematic representation of SnappyHexMesh mechanism; A: a *.stl* surface file is imported to the OpenFOAM environment from a thrid-party software; B: a 2D or 3D mesh, necessarily larger than the .stl geometry is generated by using the built-in tool *blockMesh*; C: the two former parts are fused into a mesh with the desired geometry.

Boundary conditions

After the mesh is created, the walls and in general all boundaries need to reflect the reality of the case. Boundary conditions are constraints necessary for the solution of boundary values, such as velocity and pressure. For example, in a closed cylinder acting as a pipe, the axial walls are defined as inlet and outlet, since the flow runs in the axial direction, whereas the longitudinal cylindrical wall acts as a physical barrier, containing the fluid inside.

Physical properties
The physical properties of the fluids need to be set in order to resemble the desired real fluids. In this way, the kinematic viscosity and the density needs to be defined. Also, if the problem to be solved is multiphasic, additional parameters such as the surface tension between inmiscible fluids, or the diffusivity coefficient between miscible fluids, must also be established.

Control
The dictionary named as *controlDict* controls the parameters of the simulation. The values set in this file have no physical meaning but rather control simulation variables such as the start time, time between solving steps, duration of the simulation, solver to be employed, etc.

1.2.2 Solving

OpenFOAM does not have a generic way to solve all the cases. Instead, the users must choose a specific solver for any kind of problem. The solvers are organized in categories whose names normally reflect, either the physical models that contain, or the type of problem they are designed to solve, for example: *imcompressible flow, heat transfer, multiphase, combustion, etc.* For each specific case, the solver is defined in the *controlDict* dictionary. In OpenFOAM, solvers can be used in parallel with the built-in *mpirun* tool. The parallel solving decomposes the problem mesh in different parts and each of these parts is assigned to a different core of the CPU. This parallel processing can greatly speed up the simulation time, which is limited by the available hardware and the complexity of the simulation.

1.2.3 Post-processing

OpenFOAM does not possess a graphical interface to visualize the simulated data. However, it is compatible with third-party software that allows data visualization. Among them, *ParaView* is a powerful software for data analysis and the most used in the OpenFOAM community. It has many filters and applications designed to treat and post-process the generated results.

2 | Objectives and scope of the project

The depressurization flow rate is an important variable in the production of *Quatsomes* by DELOS-SUSP, since it has been seen to affect the physicochemical properties of freshly made *Quatsomes*.

Thus, this work aims to analyze the DELOS-SUSP process at different Q by means of CFD tools. In this context, the following points are intended:

- The mixing of fluids at different flow rates during the DELOS-SUSP depressurization step, will be studied. This analysis will be carried ouy by simulating the process using OpenFOAM, a free-source CFD software. In this context the spatial and temporal distribution of the ethanol concentration around the mixing point will be studied.
- A correlation between the simulated data and experimentally produced *Quatsomes* will be attempted in order to glimpse the mechanism of formation of these vesicles.
- The DELOS-SUSP process will be incorporated into a CFD software environment. This implies the description of a detailed working protocol that includes: (1) pre-processing of the simulation (mesh generation, setup of boundary conditions, election of the solver...); (2) solving; and (3) post-processing of the results.

Regarding the scope of the project, the accuracy of the results is subjected to the refinement of the mesh and the models used. As an example, the solver employed assumes some simplifications, like considering the DELOS-SUSP process as fully incompressible and isothermal. It has to be taken into account that these simplifications are accepted in order to be able to provide some results with the present limitations of time and hardware.

Thus, this project explores for the first time the simulation of the DELOS-SUSP process for the preparation of vesicles, and aims to set the foundations for future works.

3 | Simulation case set-up of the depressurization step in the DELOS-SUSP process

The startup of an OpenFOAM simulation is not a straightforward task, specially for the newcomers to the software. In this section, a full description of the setup for the simulations presented in the results is described. From the beginning, this description includes a first election of a solver that fits the problem at hand, followed by the elaboration of a meshed domain where the simulation occurs, and finally, the definition of the boundary conditions and constant fluid properties.

3.1 Solver selection: *InterMixingFoam*

OpenFOAM has many different solvers, but only a few deal with multiphasic flows. Among the remaining solvers, just a very few are developed to work with more than two phases. The present project deals with three different components, two of them which are miscible, namely water and ethanol, and a third component that it is inmiscible with the former ones, namely the CO_2.

InterFoam is one of the most used solvers for multiphasic problems in OpenFOAM. This solver is intended to work with two incompressible, isothermal and immiscible fluids. Luckily, some years ago, a modification of *interFoam* was created to work with tri-component flows. This modification was baptized as ***interMixingFoam*** and it is the solver used in the present work.

The characteristics of this solver and the governing equations that define it are the following:

Incompressible
The differential form of the continuity equation in fluid dynamics is written as:

$$\frac{\delta \rho}{\delta t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{3.1}$$

where ρ is the fluid density, t is the time and $\nabla \cdot (\rho \boldsymbol{u})$ is the divergence of the density of the fluid multiplied by flow velocity vector. Incompressible flow means that ρ is constant, which simplifies the mass continuity equation to:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{3.2}$$

The momentum equation for this solver is described as in equation (3.3).

$$\frac{\delta \rho \boldsymbol{u}}{\delta t} + \nabla \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = -\nabla p + \nabla \cdot \rho\nu[2S] - \rho g \tag{3.3}$$

ρ is constant, so we could divide the whole equation by it and have the $-\nabla p$ term divided by the density. Instead, in this solver, ρ is maintained, so in the diffusion term we find $\rho\nu$, the density multiplied by the kinematic viscosity, which can be written as μ, the dynamic viscosity. S is defined as the symmetric part in the velocity gradient tensor. Finally, g is the gravity.

Transient
The solver allows changes in time, as seen by $\frac{\delta \rho \boldsymbol{u}}{\delta t}$, the partial derivative term in equation (3.3).

Multiphasic - VOF
As already said, *interMixingFoam* can work this multiphasic flows, where each phase is defined by its density and kinematic viscosity. For modeling multiphasic flows, OpenFOAM uses the volume of fluid (VOF) method. In this numerical method, a mean density is assumed, defined by equation (3.4), which is the ρ used in the previous equations (3.2) and (3.3).

$$\rho = \alpha\rho_1 + (1 - \alpha)\rho_2 \tag{3.4}$$

For simplification, equation (3.4) is defined for a biphasic flow, but an equivalent equation could be used for a triphasic one. The mean density of each cell is defined in this equation, which is determined by ρ_1 and ρ_2, the density of both inmiscible phases, and a scalar field, α, which weights the value of ρ (being max. $\alpha = 1$, where $\rho = \rho_1$ and min. $\alpha = 0$, where $\rho = \rho_2$).

In order to describe the movement of the two (or more) fluids and to define where the interfaces between the immiscible fluids are, a transport equation for α, equation (3.5), is needed:

$$\frac{\delta \alpha}{\delta t} + \nabla \cdot (\alpha \boldsymbol{u}) = 0 \tag{3.5}$$

Isothermal
InterMixingFoam assumes isothermal conditions so an energy equation is not required.

3.2 Mesh generation

Solving any case in OpenFOAM starts by creating a mesh in which the simulation occurs. The mesh is a group of individual cells and an integral part of the numerical solution. The mesh must satisfy certain criteria to ensure a valid, and hence accurate solutions. Thus, during any run, OpenFOAM checks that the mesh satisfies a set of validity constraints and will cease running if the constraints are not satisfied [25].

In OpenFOAM, a mesh can be created by using different tools or applications. In the case of this work, the meshes have started from surface geometry files (*.stl*) created with *Blender*. *Blender* is a free, open-source, three dimensional (3D) computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, etc.

3.2.1 Creation of the geometry

For the generation of the *.stl* file, the physical part where the depressurization of the DELOS-SUSP process occurs (Figure 3.1), was modelled. The physical

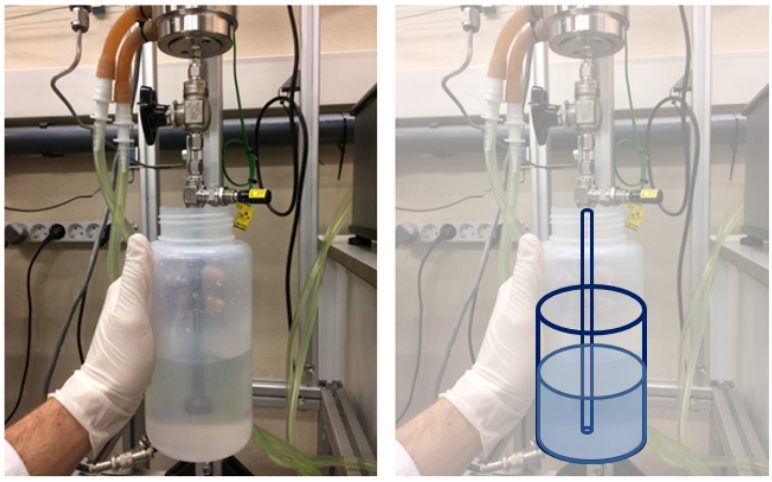

Figure. 3.1: (Left) Depressurization setup in the DELOS-SUSP process and, (right) part being modelled with blender.

elements involved in the depressurization were measured and the different elements were designed according to the original dimensions, as shown in Table 3.1:

Pipe	**Lid**	**Collector**
D = 4 mm	D = 80 mm	D = 80 mm
L = 250 mm	-	L = 11 mm

Table 3.1: Diameters (D) and lengths (L) of the *.stl* geometries for the creation of the mesh.

Instead of one single geometry that included every surface, the depressurization setup was split in three different parts (as illustrated in Figure 3.2): (1) the pipe, which is the metal tube going from the depressurization valve until the aqueous phase; (2) the collector, the vessel containing the aqueous phase where the organic phase is depressurized; and (3) the lid, which does not exist physically, but acts as the open end of the collector. Splitting the geometry in three items eases the setting of the BCs, topic that will be covered later on this thesis.

Also, it must be noted that for simplicity, all the walls of the previously stated geometries are two-dimensional manifolds, or in other words, they have no thickness.

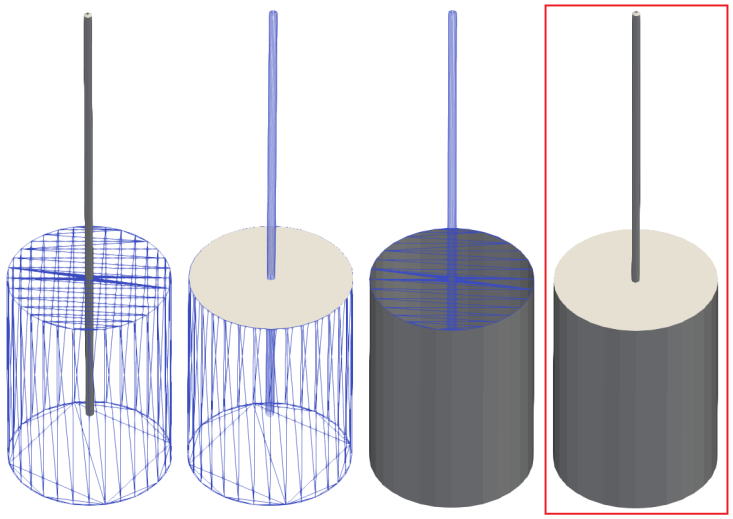

Figure. 3.2: Blender files for (from left to right) the pipe, lid and collector. The final file with all geometries together is framed in red.

3.2.2 Meshing of the geometry

Once the desired geometry is created, the following step is to generate a 3D mesh in order to fuse it with the geometry into one single element.

For this reason, a hexahedral mesh was generated with an application called *blockMesh*. This application is a basic yet powerful mesh generator capable of creating volumes made of hexes. The parameters of a mesh created with *blockMesh* are controlled by a dictionary called *blockMeshDict*. Information regarding the structure of this dictionary can be found in appendix B: *Dictionaries*.

A representation of said hexahedral mesh can be seen in Figure 3.3. The number of cells of the mesh is an important parameter that influences the running time needed for completing the simulation. Thus, a coarse mesh (with bigger cells) will yield results faster than a finer one (with smaller cells). As a counterpart, the accuracy of the results will also be determined by the quality of the mesh, the finer the mesh, the more accurate will be the data retrieved.

Once the geometry and the hexahedral mesh were generated, the meshed geometry was produced using *snappyHexMesh*. This application chisels the mesh into the desired shape using as a template an already existing surface like the one presented in Figure 3.2. *SnappyHexMesh* requires a series of commands in order to generate a good quality mesh. These guidelines are specified in the *snappyHexMeshDict* dictionary (Appendix B: *Dictionaries*).

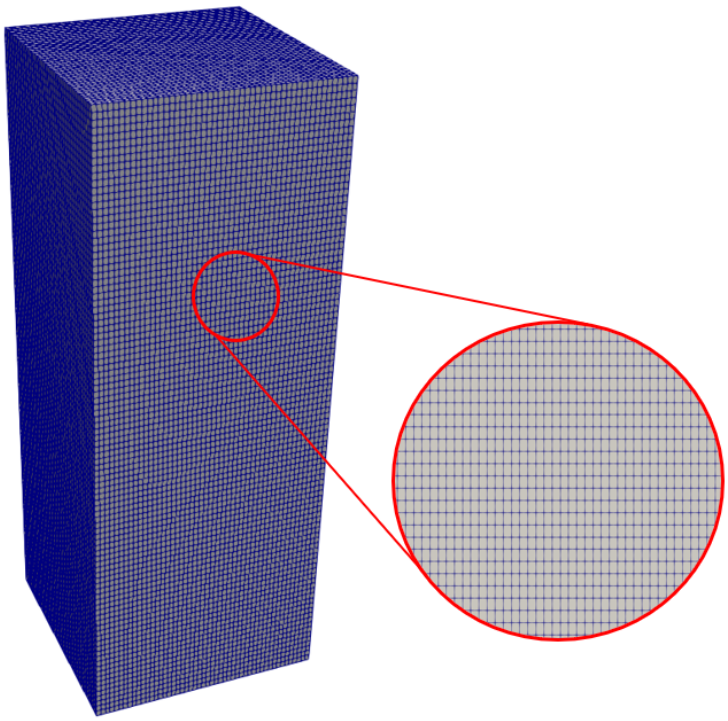

Figure. 3.3: Hexahedral mesh created with *blockMesh*.

Also it must be noted that the depressurization step of DELOS-SUSP can be considered symmetric in the two directions perpendicular to the longitudinal axis of the depressurization pipe, as schematized in Figure 3.4. As a result of this consideration, it was decided to simplify the problem and mesh only one fourth of the geometry with *snappyHexMesh*, as shown in Figure 3.5. Decreasing the number of cells of the problem to a quarter of the original would reduce the running time of the simulations proportionally, making the development of the project more time-efficient while still being representative.

As can be seen in Figure 3.5, the central part of the mesh (where the pipe is located) has a higher level of refinement which means that the cells around this area are smaller. The volume embedded inside the pipe consists on a cylinder with a diameter of 4 mm. For a cylinder with such a small cross-sectional area, it was necessary to increase the number of cells in order to differentiate the cells belonging to the wall of the pipe —which are constraint by the boundary conditions—, to the cells placed in the center of the pipe —which are free from those constraints—. This problem is further explained in appendix A.1: *Troubleshooting*.

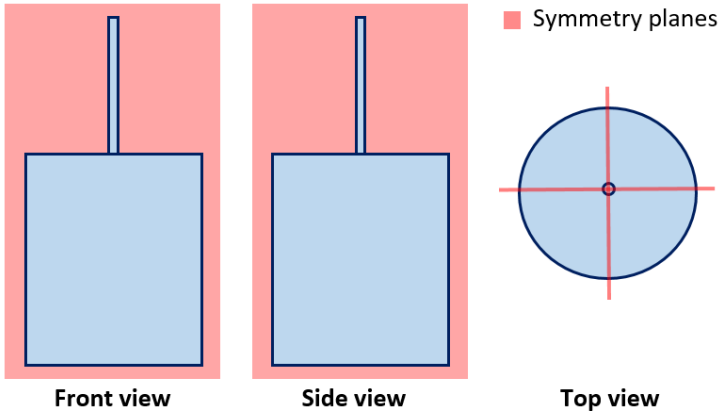

Figure. 3.4: Front, top and side view of a schematization of the depressurization setup.

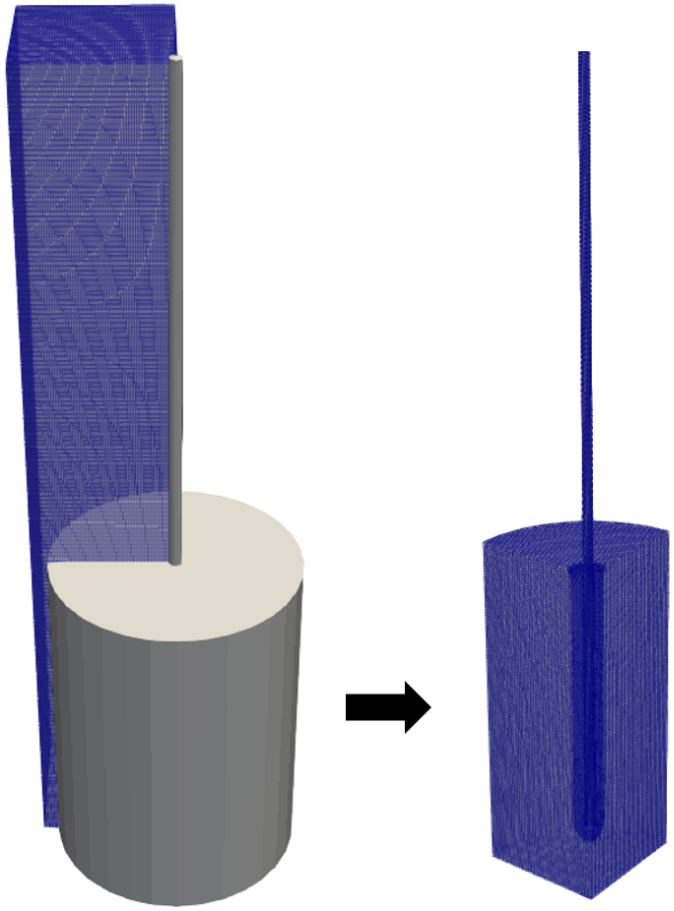

Figure. 3.5: Meshed geometry from Figure 3.2 and hexahedral mesh from Figure 3.3 are combined together with *snappyHexMesh* to form a mesh with the desired shape.

3.3 Boundary conditions and initial cell values

3.3.1 Set-up of boundary conditions

As previously stated, BCs are defined as constraints necessary for the solution of a boundary value. OpenFOAM libraries contain hundreds of different BCs, and the selection of every BC must be thoroughly analyzed, since they will define how the flow will behave in the boundaries, and by extension, in the whole meshed domain. The mesh present in this work was split in 6 parts in order to better define the BCs (Figure. 3.6).

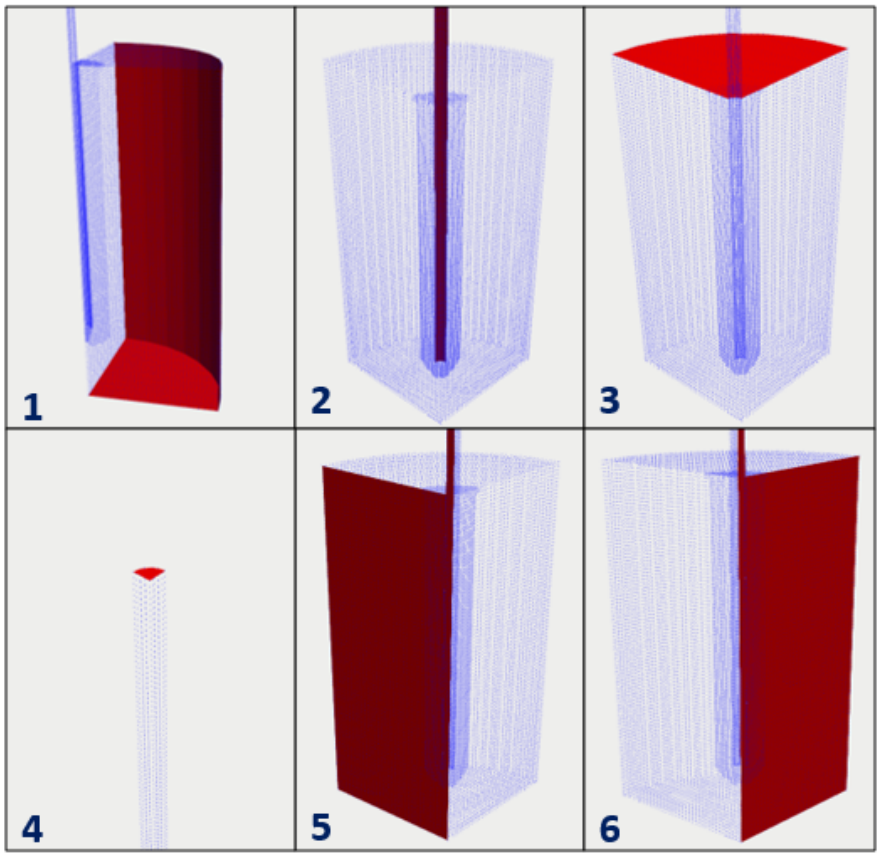

Figure. 3.6: Mesh splitted in 6 patches for defining boundary conditions: 1 - collector; 2 - pipe; 3 - lid; 4 - inlet; 5 - symmetry plane A; 6 - simmetry plane B.

OpenFOAM solves the values of pressure and velocity of the flow for every cell in every time step of the simulation. Thus, BCs are needed for the solution of P and U for every one of the aforementioned patches. A

compilation of the BCs used for the present project is presented in Table 3.2, followed by a brief explanation for each term.

Mesh section	BCs for Pressure	BCs for velocity
Collector	*fixedFluxPressure*	*noSlip*
Pipe mm	*fixedFluxPressure*	*noSlip*
Lid	*totalPressure*	*pIOVelocity*
Inlet	*fixedFluxPressure*	*fixedValue*
Symmetry P. A	*symmetryPlane*	*symmetryPlane*
Symmetry P. B	*symmetryPlane*	*symmetryPlane*

Table 3.2: Compilation of the boundary conditions for P and U.

fixedFluxPressure: Sets a gradient of pressure in the cell so that the flux on the boundary is the one specified by the velocity boundary condition.

totalPressure: Sets a pressure equivalent to the atmospheric one.

noSlip: Fixes velocity $U = 0$ m/s for the specified walls

pIOVelocity: Or without abbreviation, *pressureInletOutletVelocity*. This boundary condition provides a combined effect. For flow pointing outwards of the domain, it sets the gradient of velocity to 0, so the velocity at the boundary is the same as at the center of the last cell. For inflows, assigns a velocity to the flow based on the flux perpendicular to the patch where this BC is set.

fixedValue: Fixes a specific value (defined by the user) for the velocity.

symmetryPlane: A plane that mirrors the pressure and velocity that has in front. This BC is the one that allows this problem to be simplified due to its symmetry as schematized in Figure 3.4.

3.3.2 Initial cell values and constant properties

Along with the BCs, the initial conditions from where the simulation starts had to be defined. Velocity, pressure and composition of every cell are defined in different OpenFOAM dictionaries in the *0* folder (see Figure 1.5). Detailed information regarding this dictionaries can be read in appendix B: *Dictionaries*.

The velocity of the flow was set as $U = 0$ m/s for almost all the meshed domain for $t = 0$ s, implying that initially, there was not movement of the fluids. As an exception, the only section with a fixed value for the velocity was the inlet, where the velocity was set according to the desired flow rate for that run. Therefore, the velocity in the inlet was calculated as Q (the values of Q have been obtained empirically (see section 4.1, equations (4.3) and (4.4)) divided by the cross-section area of the depressurization pipe.

$$\boldsymbol{Pipe\ cross\ section} = \pi \cdot r^2 = \pi \cdot 0.002^2 = 1.25\ \cdot 10^{-5}\ m^2 \tag{3.6}$$

$$For\ low\ Q\ \rightarrow \boldsymbol{U} = \frac{35\ \cdot 10^{-6}\ m^3/s}{1.25\ \cdot 10^{-5}\ m^2} = 2.8\ m/s \tag{3.7}$$

$$For\ high\ Q\ \rightarrow \boldsymbol{U} = \frac{175\ \cdot 10^{-6}\ m^3/s}{1.25\ \cdot 10^{-5}\ m^2} = 14\ m/s \tag{3.8}$$

For the pressure, the pipe and the collector were set at atmospheric pressure. Again, the only section with a different pressure value was the inlet. Opposite to the velocity, that was defined by the user, the pressure was automatically set by OpenFOAM in order to achieve the desired velocity in the Inlet.

Finally, for the composition, every cell was set with a certain volume fraction of the components (i.e., water, ethanol and CO_2). Specifically, the cells in the lower 6.5 cm of the collector were set as water. The rest of the collector and the inside of the pipe were set as CO_2. Ideally, the collector should had been filled with water and atmospheric air. However, including air meant adding a fourth component that would have required the use of an alternative, much more complicated, type of solver. Thus, for this project, it was considered that the thermodynamic properties of CO_2 were similar enough to air as a proof of concept. Ethanol (EtOH), was not present in the simulation domain at $t = 0$ s except in the inlet. A constant condition was set in the inlet in order to specify the composition of the flow. As so, the volume fraction of ethanol in the inlet cells was calculated from the empirical volumes used to perform the DELOS-SUSP (see Section 4.1):

$$\boldsymbol{Volume\ frac\ CO_2\ inlet} = \frac{20000\ cm^3\ CO_2}{20000\ cm^3\ CO_2 + 31.5 cm^3\ EtOH} = 0.998 \tag{3.9}$$

Thus, the composition of the inlet flow was defined as 99.8% CO_2 and 0.2% EtOH. This composition was maintained constant throughout the simulation.

Regarding the flow composition, OpenFOAM does not possess a library with a list of pre-defined components ready to be selected. Instead, these components are specified by fixing some properties that are characteristic for the fluids for a given thermodynamic state. Also, as previously mentioned, *interMixingFoam* allows the co-existence of three phases, two miscible and a third one immiscible with the first ones. The diffusion coefficient between miscible components and the surface tension between immiscible ones were also established to define the mixture. Table 3.3 summarizes the constant properties of the fluids.

Component	**Density** (Kg/m^3)	**Kinematic viscosity** (m^2/s)	**Diffusion coeff.** (m^2/s)	**Surface tension coeff.** (N/m)
Water(1)	998	$9.56x10^{-7}$	-	0.021(1-3)
Ethanol(2)	789	$1.52x10^{-6}$	$0.871x10^{-9}$(2-1)	0.072(2-3)
CO_2(3)	1.82	$8.05x10^{-6}$	-	-

Table 3.3: Constant properties of the simulation components. All densities and kinematic viscosities were obtained from the NIST [26]. Diffusion coefficient of ethanol in water (2-3) was obtained from [27]. Surface tensions coefficients (1-3) and (2-3) were obtained from [28].

4 | Results and discussion

4.1 Experimental results: preparation of *Quatsomes* at different depressurization flow rates

Previous works had already foreseen a possible correlation between the homogeneity of the *Quatsomes* and the depressurization flow rate, Q, in their production. However, a study dedicated to prove this theory had never been carried out before. Hence, to prove this hypothesis, several experiments were perfomed in a 75 mL DELOS-SUSP high pressure plant.

In those experiments, samples were produced as described in Section 5.3. For the preparation, 90 mg of cholesterol were solubilized in 31.5 mL of ethanol at 308 K. The solution was then introduced in the autoclave along with 39.5 mL of compressed CO_2. The volume of CO_2 introduced was measured with the syringe pump that injected the compressed gas into the reactor (at P = 10 MPa, T = 272 K). The quantities and volumes aforementioned were meant to keep approximately $X_{CO2} = 0.62$, a molar fraction in which is known that all the components are solubilized at the chosen P and T. After one hour at working conditions, (P_W = 10 MPa, T_W = 308 K), the CO_2-expanded organic solution was depressurized over 285 mL of ultra pure water.

This experiments were performed using two different Q. The depressurization flow rates were controlled manually, by further opening or closing the micrometric depressurization valve for shorter or longer times, namely 2 and 10 minutes. Each experiment was reproduced by triplicate to check for deviations in order to validate the empirical results.

For calculating Q, a thermocouple was placed as close as possible (2 cm) after the depressurization valve *V-8* (see Figure 5.2). Measuring the temperature after the outlet was crucial for determining Q since the density of compressed fluids is greatly dependant on its temperature and pressure. In the DELOS-SUSP, the CO_2 is being depressurized from 10 MPa to atmospheric pressure. This abrupt decrease in the pressure produces the evaporation of the CO_2, causing a sharp reduction of the fluid temperature. On average, the temperature registered after the depressurization was T = 285 K. With the empiric values for P and T registered in the depressurization, the CO_2

density at the 2 different thermodynamic conditions was extracted from the *NIST: National Institute of Standards and Technology* [26].

	10 MPa, 272 K	**0.1 MPa, 285 K**
CO_2 density (kg/m^3)	988.74	1.88

Table 4.1: Carbon dioxide density at different thermodynamic states.

The volumetric flow rate for each case was determined as follows:

$$Q = \frac{Total\ volume}{Depressurization\ time} \tag{4.1}$$

The total volume being depressurized comprised the organic liquid phase, in this case the EtOH solution, and the expanded gas phase, the CO_2. The volume of CO_2 (gas) being released in every experiment can be calculated as:

$$39.5\ mL\ CO_2\ \cdot \frac{1\ m^3}{10^6\ mL} \cdot \frac{988.74\ kg}{1\ m^3} \cdot \frac{1\ m^3}{1.88\ kg} = 0.02\ m^3 = 20000\ cm^3\ CO_2 \tag{4.2}$$

For the two selected depressurization times, 2 and 10 minutes, Q was calculated as:

$$Q_{low} = \frac{20.000\ cm^3\ CO_2 + 31.5\ cm^3\ EtOH}{600\ s} \sim 35\ cm^3/s \tag{4.3}$$

$$Q_{high} = \frac{20.000\ cm^3\ CO_2 + 31.5\ cm^3\ EtOH}{120\ s} \sim 175\ cm^3/s \tag{4.4}$$

Results from equations (4.3) and (4.4) does not yield exactly 35 and 175 respectively. However, since what is important is the difference between them and not the exact value, we will work with this natural numbers from now on, in order not to drag decimal numbers.

After the preparation of the *Quatsomes*, they were characterized over a period of time to check the evolution of its characteristics. Figure 4.1 shows the evolution of size and polydispersity index (PdI) of the vesicles, respectively over 4 weeks. The analysis was performed by employing the DLS technique.

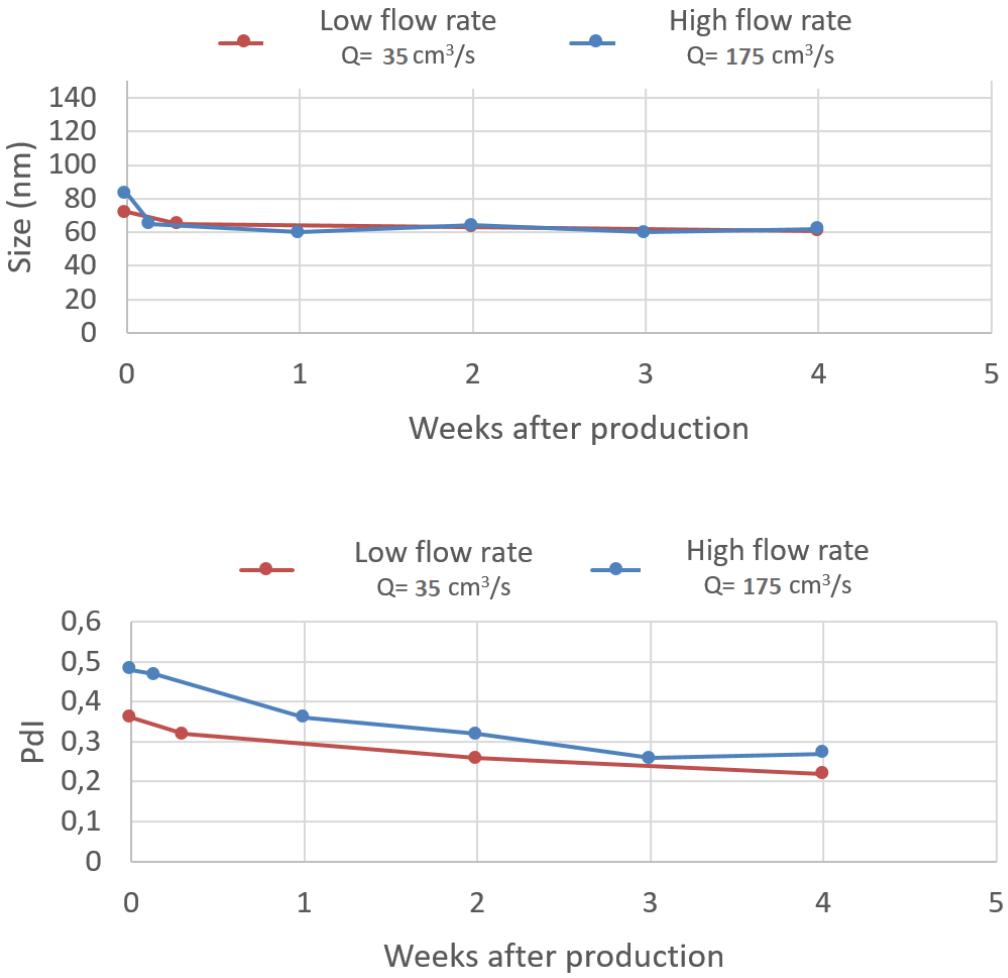

Figure. 4.1: Average size and PdI comparison between *Quatsomes* produced at low and high flow rates.

From the DLS results, it can be seen that, even if the PdI is stabilized over time, the initial value is notably higher in the case of higher Qs. For visual support and verification of the DLS results, cryo-TEM representative images were taken from both Q configurations. The images were taken 1 week after the production of the vesicles. Weekly analysis using electronic microscopy were not possible due to the high economic costs of this technology.

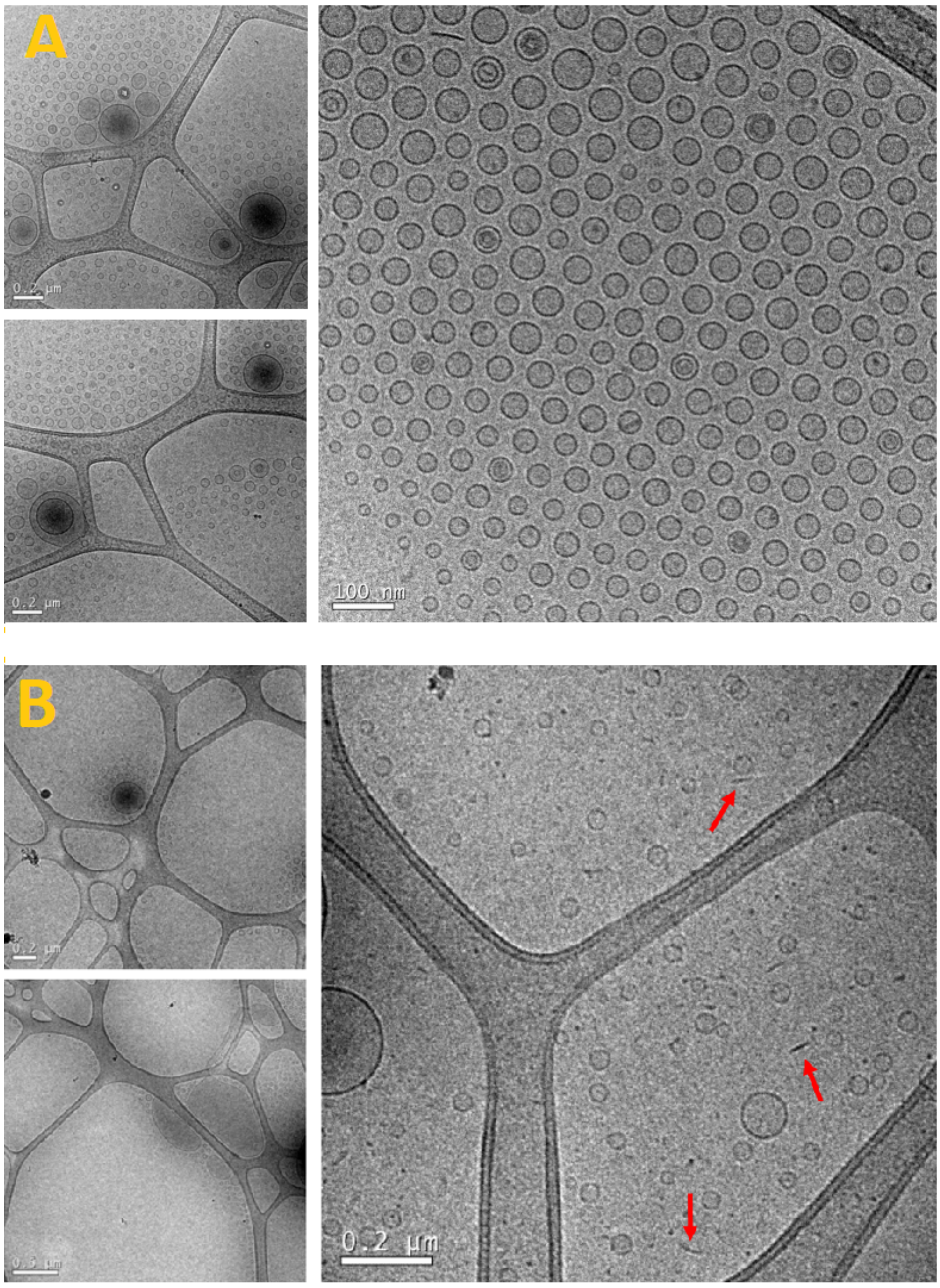

Figure. 4.2: cryo-TEM images of Quatsomes produced at (A) low (Q = 35 cm^3/s) and (B) high flow rate (Q = 175 cm^3/s). 1 week after vesicles production.

As shown in Figure 4.2, after the first week of preparation, there is a clear difference in terms of quantity and homogeneity between the *Quatsomes* prepared at different flow rates. Vesicles prepared at low Q are abundant and show a high homogeneity in size and shape. On the other hand, vesicles prepared at high Q, are scarcer and more heterogeneous. In this case, the absence of sphere-shaped vesicles is sometimes replaced with stick-shaped or rod-shaped structures (pointed in Figure 4.2 B with red arrows), which can not be assumed as viable DDs.

4.2 Simulation of the depressurization step at different flow rates

In this section, the results of the simulations of the depressurization step of DELOS-SUSP at 35 cm^3/s and 175 cm^3/s are presented. Also, a correlation between the physicochemical properties of *Quatsomes*, and the results from the simulations is intended.

Before the goals of this project were defined, the initial objective was to include into the simulations the building molecules of the *Quatsomes*, namely the CTAB and the cholesterol. The idea was to observe the temporal and spatial distribution of these two species for different chosen flow rates and extract conclusions based on those distributions. After research, the idea was removed from the scope of the project due to the huge complexity that involved adding two new species to the already multiphasic flow. Instead, as a step towards the characterization of the system by CFD, **the distribution of the depressurized ethanol was analyzed**, since the cholesterol is assumed to be homogeneously dissolved in the ethanol.

Thus, the concentration of ethanol in space and time was studied by sampling the volume fraction of ethanol at different heights and radii throughout the collector. The analysis was made over all the circular lines defined by the chosen radii, for each height. A schematization of the "sampling lines" can be seen in Figure 4.3.

Ideally, the simulations at the two different flow rates should have been simulated for 2 and 10 minutes to fully represent the experiments shown in section 4.1. However the results presented below were run for only 15 seconds. The reasons of such short simulation times were the numerous convergence problems appearing at longer times and the high time consumption of the

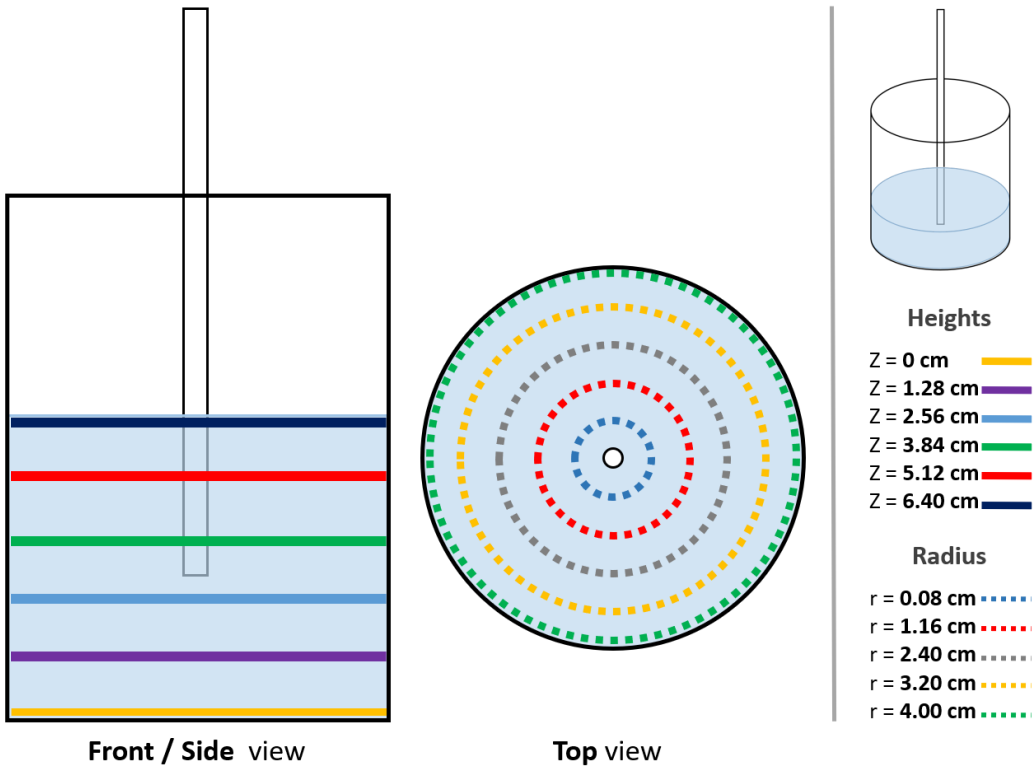

Figure. 4.3: Schematic representation of the sampling spots of the domain. The volume fraction of ethanol was sampled over circular lines at 5 different radii over 6 heights, making a total of 30 sampling spots for each Q.

simulation.

Specifically, **every second in the simulation took up to an average of 14.2 hours of processing time**. This running time was the compromise between using a fine enough mesh but still obtaining results in a reasonable time. The simulations were run in parallel using 8 cores of the cluster described in Section 5.1. The long processing times are caused by the multiphasic nature of the problem, which is extremely expensive in terms of processing power.

Hereunder, the results from the simulations at 35 cm^3/s and 175 cm^3/s are shown. More detailed and enlarged plots of these same results can be found in Appendix C.1.

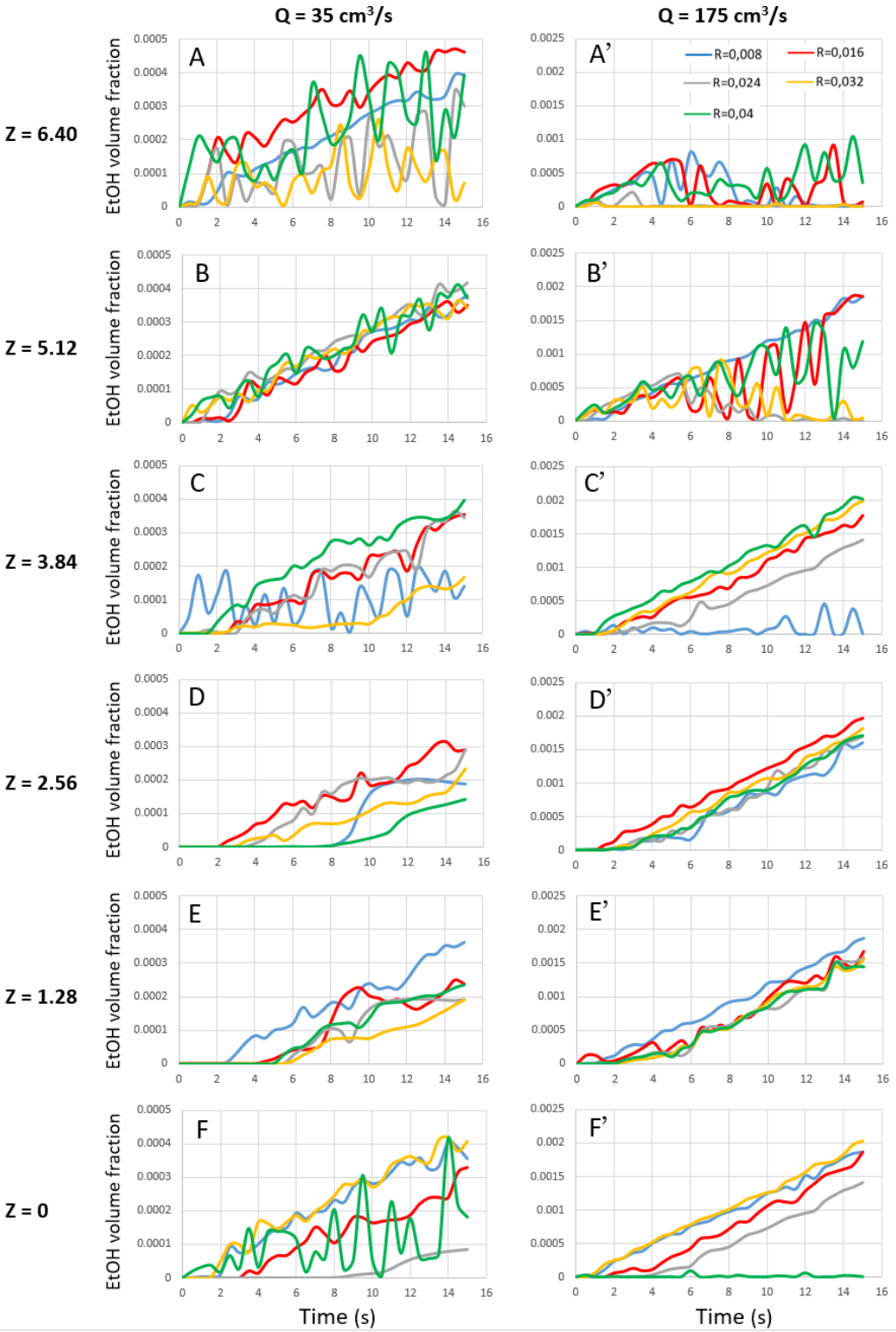

Figure. 4.4: Ethanol volume fraction *vs.* time in the collector while the depressurization of the organic phase occurs. The charts are labeled from A to F'. The plots are separated in 2 columns for the 2 different depressurization flow rates and in 6 rows for every height. The sampling is made at different radii differentiated by color lines inside each chart.

The first thing to notice above the previous charts is that they are only half consistent in terms of total ethanol concentration. One one hand, the volume fraction in these simulations is inferior to what would be expected from the experiments:

The volume fraction of ethanol in the water at the end of a experiment is:

$$\frac{31.5\ mL\ EtOH}{31.5\ mL\ EtOH + 285\ mL\ water} = 0.099 \sim 0.1 \tag{4.5}$$

As explained in section 4.1, $Q = 35$ cm^3/s, comes from depressurizing the whole content from the pressurized vessel (the organic phase containing compressed CO_2 and ethanol) in 10 minutes (600 s), while $Q = 175$ cm^3/s is the flow rate obtained when depressurizing the same amount of volume in 2 minutes (120 s). Then, 15 seconds of simulation, is just a 2.5% of the process for the higher Q, and a 12.5% of the process for the lower Q. Seeing that the volume fractions are in the range 0.0005 and 0.0025 (see Figure 4.4), the volume fraction at the end of the simulations would be:

$$For\ Q = 35\ cm^3/s \rightarrow \frac{100\%}{2.5\%} \cdot 0.0005 = 0.02 \tag{4.6}$$

$$For\ Q = 175\ cm^3/s \rightarrow \frac{100\%}{12.5\%} \cdot 0.0025 = 0.02 \tag{4.7}$$

Naturally, a volume fraction of 0.02 is inferior to 0.1. A possible explanation to this difference from the expected value could be that the velocity in the inlet (and as a consequence, the flow) is inferior due to a poor refinement of the mesh. However, one of the main drawbacks of this project has been the very long processing times of the simulations, preventing a better refinement of the mesh. This problematic regarding inlet velocity due to a poor refinement of the mesh is widely explained in Appendix : A.1.

However, on the other hand, the results of the simulations are consistent between themselves. The process has been simulated at $Q = 35$ cm^3/s and $Q = 175$ cm^3/s, which is a Q five times superior to the former value. In turn, all the presented plots for $Q = 35$ cm^3/s, fit in the volume fraction range of 0.0005 while the plots for $Q = 175$ cm^3/s fit nicely in the range of 0.0025,

which is again 5 times superior.

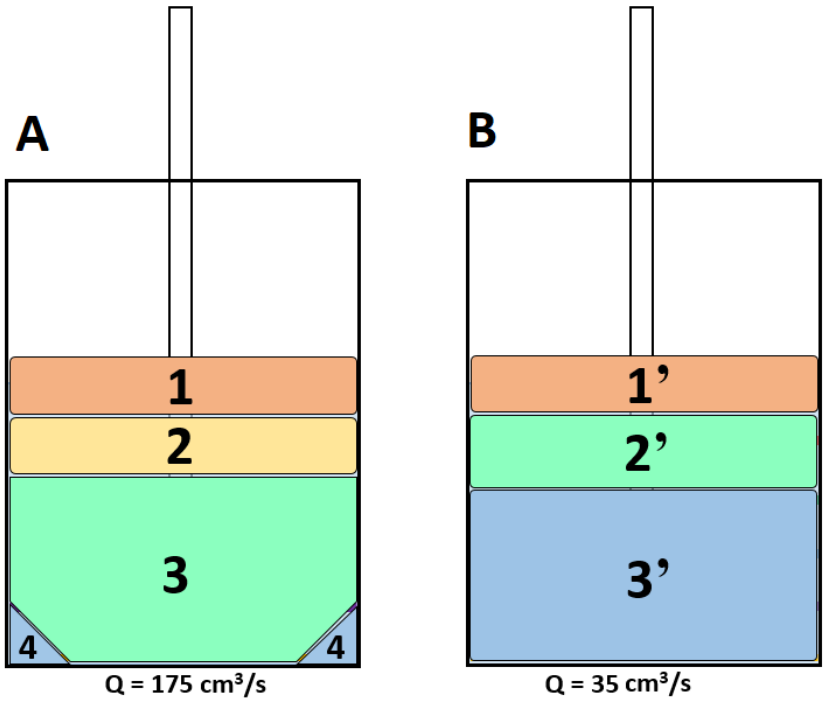

Figure. 4.5: Different distributions of the ethanol volume fraction delimited by color zones.

After analysis of the plotted results, Figure 4.5 tries to schematize the information gathered from the graphs in Figure 4.4:

Looking first at Figure 4.5: *A*, for $Q = 175$ cm^3/s, one can notice that there are 4 different colour zones, each one of them numerated. Zone 3 represents the lowest part of the collector. In this area, the volume fraction of ethanol increases lineally as could be expected from ethanol mixing into water upon the vigorous agitation produced by the CO_2 bubbling. This lineal trend can be observed in plots 4.4: *C', D', E' and F'* for almost all radii. As exception, in the latter, for R = 4 cm (the furthest from the center), there is no increase in the volume fraction of ethanol, that could be attributed to a poor mixing in the bottom corners (Zone 4).

Zone 1 (orange), represents the results plotted in chart 4.4: *A'* . A more or less irregular increase of the volume fraction can be seen for almost all the radii until t = 2 - 3 s, followed by a drop to almost 0 beyond that time. This drop can be explained because the patch of the mesh formerly denominated as "lid" in Figure 3.6 (the open end of the collector), is programmed to prevent

back flows. This open end is set as a flow outlet to avoid the pressurization of the collector due to the injection of the high volume of CO_2. As a counterpart, the liquid that is ejected out of the meshed domain due to the vigorous bubbling that is produced at the high Q (175 cm^3/s), does not return to the domain. As a result, the collector is emptied little by little as can be seen in Figure 4.6. In the real-life experiments, the emptying of the collector is prevented by placing a perforated film on top of the open end, allowing the CO_2 to escape but retaining the liquid. In the simulations, setting a BC which is capable of imitating this perforated film is something very complex and not contemplated in the scope of this project.

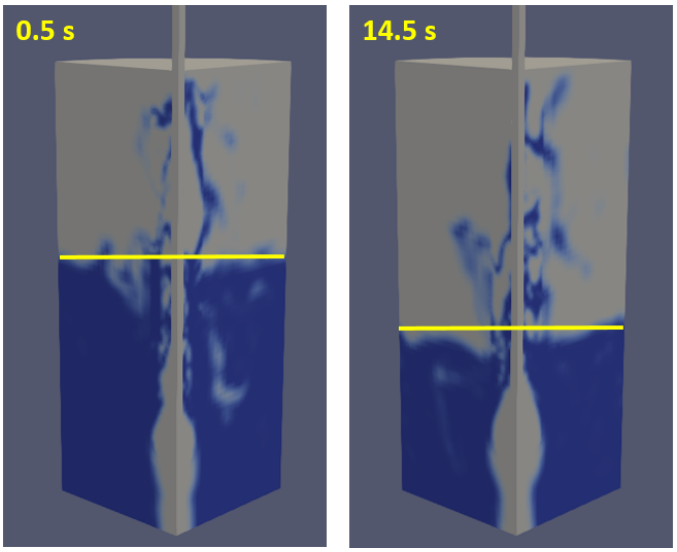

Figure. 4.6: Liquid level difference at the beginning and at the end of the simulation at high Q.

Finally, zone 2 (yellow) in Figure 4.5 : *A*, referring to chart 4.4: *B'*, follows the same trend as zone 1, but the decay starts later, at t = 5 - 6 seconds, since the sampling is made in a lower point and it takes more time to empty this volume.

On the other hand, Figure 4.5: *B* summarizes the fluids distribution for the simulation at low flow rate, $Q = 35$ cm^3/s. In here we can distinguish 3 differentiated behaviours delimited by the three colored zones.

In zone 2' (green), representing the data from plot 4.4: *B*, a more or less lineal increase of the volume fraction of the ethanol can be observed. In this scenario, there is also CO_2 bubbling, but the agitation produced by it, is far less vigorous than at high flow rate, so it is assumed than the main form of transport for the ethanol is by diffusion. Thus, the lineal trend in this

area could be explained as the ethanol, since it is less dense than water, it is slowly diffusing upwards.

The behaviour in zone 1' (orange) would be expected to have a similar behaviour than zone 2', since it is in the upper part of the collector. However, upon observation of chart 4.4: *A*, a high disparity of the volume fraction of ethanol in time can be seen. This disparity has been attributed to the displacement of the liquid - atmosphere interface. The data in this plot is obtained by sampling the ethanol at almost the highest part of the aqueous phase (see Figure 4.3, front view, dark blue line). The bubbling produced by the CO_2 that is injected into the liquid bulk, randomly agitates this interface causing the system to detect an increasing concentration of ethanol when the liquid rises in the sampling lines, and a decrease of the concentration when the liquid level goes down (since in that moment the "probe" is mostly detecting atmospheric gas)

Finally, zone 3' (blue) represents the behaviour at the lowest parts of the collector. This region is defined by the graphs in figures 4.4: *C, D, E* and *F*. The volume fraction in here also seems to increase in a more or less lineal way but a clear conclusion can not be extracted from the plotted data, since it shows much more disparity than its high Q counterparts. A possible explanation could be the formation of moving ethanol clusters in this zone, producing sharp increases and decreases of the ethanol volume fraction in space and time. Another explanation could be that the current mesh in which the simulations have been performed is not fine enough to resolve the small ethanol concentration at the bottom of the collector for this Q.

4.2.1 Comparison of the ethanol distribution at different flow rates

Even if the trends observed in Figure 4.4 can provide some hypothesis, the results are not directly comparable. Due to hardware and time resources available, only 15 seconds for each flow rate were simulated, which as already said, mean a 2.5% for the simulations at low Q and a 12.5% for the ones at high Q. Ideally, in order to compare the results at different flow rates, the process should be at the same "stage".

Thus, in order to achieve a direct comparison, **the abscissa axis in those graphs was normalized by dividing the time of the simulation (t) by the total time that the process would take at each flow rate (t_f)**, namely 600 seconds for the simulations at $Q = 35$ cm^3/s and 120 seconds for the ones at $Q = 175$ cm^3/s. The new values (t / t_f) were then multiplied by 100 in order to obtain the completed percentage of the process. This analysis was performed only for the four plots in the lower end of the collector. The results for heights Z = 5.12 cm and Z = 6.40 cm were determined as non comparables due to the problem of the collector being emptied at this height in the high Q simulations.

Hereunder, the normalized results of ethanol volume fraction *vs.* completed percentage of the process are shown:

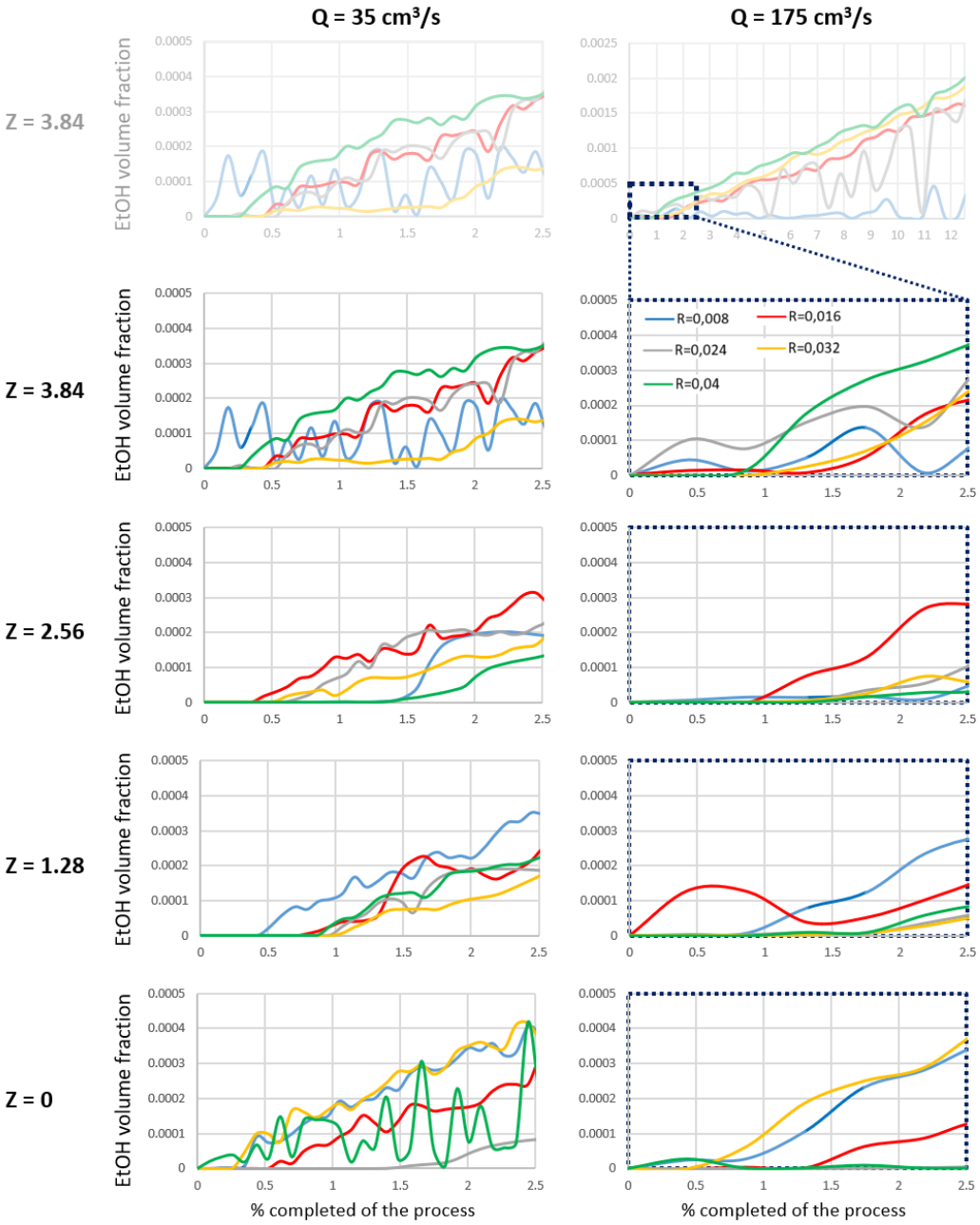

Figure. 4.7: Ethanol volume fraction *vs.* % completed of the process in the collector while the depressurization of the organic phase occurs. The plots for $Q = 175$ cm^3/s are zoomed in the first 2.5 % in order to compare. The plots are separated in 2 columns for the 2 different depressurization flow rates and in 4 rows for every height analyzed. The sampling is made at different radii differentiated by color lines inside each chart.

The charts shown in Figure 4.7 present the results normalized by t_f. The percentage of the process completed with 15 seconds of simulation for $Q = 175$ cm^3/s is 12.5%, however, the charts are zoomed in the first 2.5% in order to be at the same stage as in the simulations at low Q.

Looking at the normalized results, for the simulations at $Q = 35$ cm^3/s, the volume fraction of ethanol seems to increase faster in most of the radii and heights than for the ones at $Q = 175$ cm^3/s. However, no general conclusions regarding how the depressurization flow rate affects the physicochemical properties of the *Quatsomes* can be obtained. The data is being compared at the 2.5% of process, a very early stage of the depressurization in which probably, a possible stationary state has not been achieved. A proper comparison would need a more advanced simulation in which a trend in the ethanol distribution could be correlated with the morphology of the *Quatsomes*. Nonetheless, these results are a first step for future, more extensive research in the characterization of the DELOS-SUSP process by computational fluid dynamics.

5 | Equipment and experimental part

5.1 Software and devices for the simulations

The depressurization step of DELOS-SUSP process was simulated in OpenFOAM v1812 provided by the ESI-group from OpenFOAM.com; the geometries for the mesh were created with Blender v2.80 and the data from the simulations was treated with ParaView v5.4.1; all the software were run in a Linux OS: Ubuntu v19.04. Regarding the hardware used, preliminar tests were performed in a Lenovo Thinkpad intel core i7, of 4 cores, and the full simulations were run in a Flexicast server, placed in a cluster formed by 2 nodes of 2 AMD Opteron processors, each one of 16 cores and with 64 GB RAM, owned by Centre LABSON , and located at the UPC-ESEIAAT.

5.2 Materials for *Quatsomes* preparation

5-Cholesten-3β-ol (cholesterol, purity 95%) was obtained from Panreac (Barcelona, Spain). CTAB, ultra purity for molecular biology, was purchased from Fluka-Aldrich. Ethanol, HPLC grade, was obtained form Teknokroma (Sant Cugar del Vallès, Spain) and CO_2 was supplied by Carburos Metálicos S.A. (Barcelona, Spain). The water used for the production for the vesicles was pretreated with the Milli-Q Advantage A10 water purification system (Millipore Ibérica, Madrid, Spain).

5.3 Preparation of *Quatsomes* by DELOS-SUSP

Equipment: The preparation of *Quatsomes* by DELOS-SUSP was performed in a 75 mL lab-scale plant schematized in Figure 5.2. A list of its

components can be found in Table 5.1 The configuration comprises a 75 mL vessel (Parker Autoclave Engineers S.A. Ohio, US) (Figure 5.1), (R), whose temperature is controlled by the use of a built-in heating jacket; a thermostated syringe pump (model 260D, ISCO inc., Lincoln, US)(PCO_2) is used to introduce the CO_2 inside (R) through valve *V-5* and *V-6*; a depressurization valve, *V-8*, which follows an on/off valve, *V-7*, from which the expanded liquid solution is depressurized into the aqueous phase placed in a collector (C); N_2 is introduced through *V-2* and *V-3* from a pressurized reservoir; both, the N_2 and CO_2 lines have non-return valves (*NRV-1*, *NRV-2*), to prevent back flows and avoid contamination in the lines; attached to R, there is an analogical pressure gauge (PI), and a digital pressure indicator controller (PIC); a thermocouple is connected to a temperature controller (TI); the sample inside the autoclave is stirred by means of a digitally controlled shaker *(VF)*. Finally, for safety reasons, there is a pressure rupture disk (DR), calibrated at a maximum pressure of 20 MPa, to avoid overpressure of the equipment in case of device malfunctions or human error.

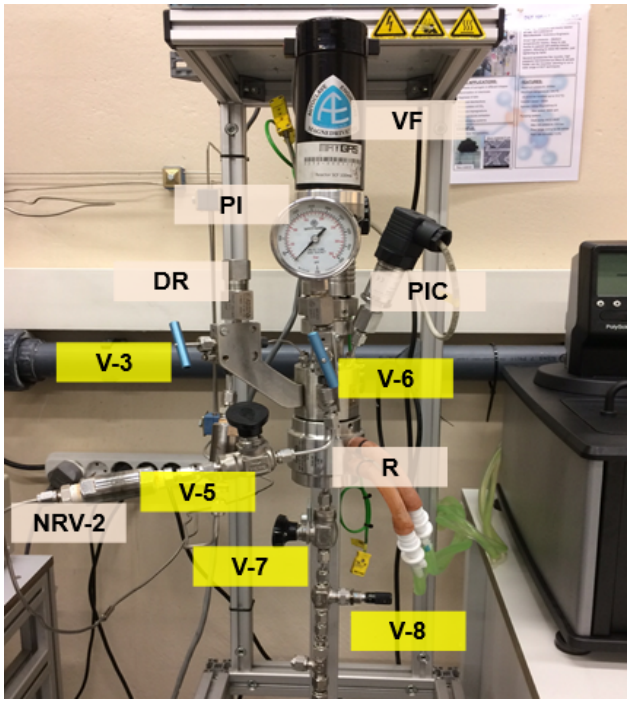

Figure. 5.1: Main parts of the 75 mL high pressure plant used for the production of vesicles.

Experimental procedure: The preparation of *Quatsomes* by DELOS-SUSP was performed according to the following procedure. A volume V

of a solution of cholesterol in EtOH is introduced into the autoclave, which has been previously brought to the working temperature (T_W = 308 K). After 10 minutes, once the solution has reached T_W, the autoclave is pressurized with compressed CO_2 through valve *V-6*, producing a volumetric expanded liquid solution with the desired molar fraction of CO_2, X_{CO2}, and at the working pressure P_W, typically 10 MPa. The organic mixture (cholesterol/EtOH/CO_2) is left to homogenize under vigorous stirring during at least one hour. Afterwards, the CO_2-expanded solution is depressurized from P_W to atmospheric pressure through valves *V-7* and *V-8* over an aqueous solution containing the surfactant molecules (CTAB). A flow of N_2 at P_W is used as plunger to push down the expanded solution in order to maintain constat the pressure inside the vessel during all the depressurization process.

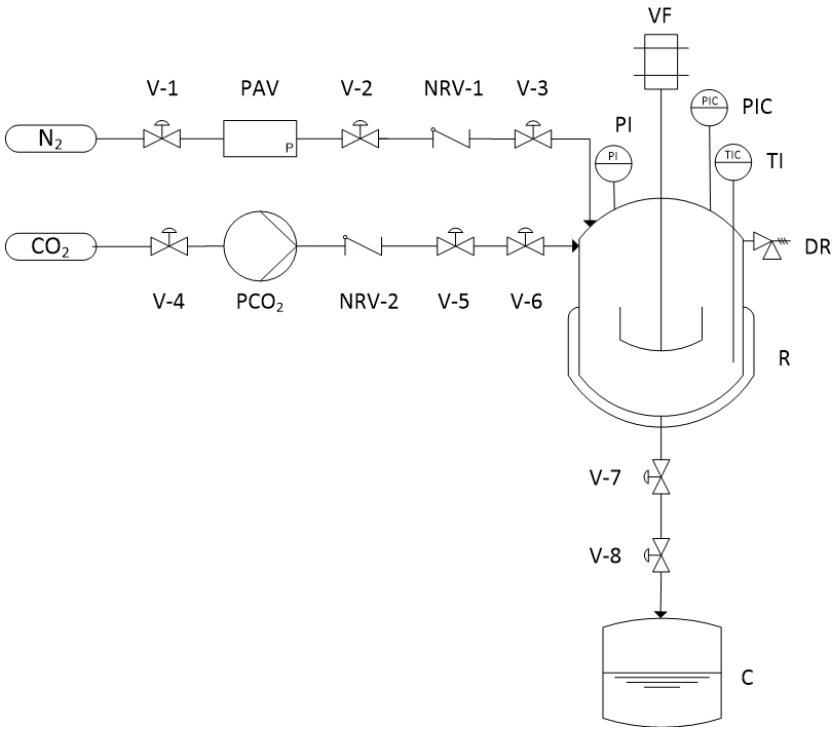

Figure. 5.2: Process flow diagram of the 75 mL plant for vesicles production.

Code	Description	Code	Description
V-1	N_2 outdoor supply valve	R	Vessel or autoclave
PAV	Pressure adjustment valve	VF	Stirrer
V-2	N_2 supply valve to plant	PI	Pressure indicator
NRV-1	Non-return valve for N_2	PIC	Pressure indicator cont.
V-3	N_2 supply valve to vessel	TI	Temperature controller
V-4	CO_2 outdoor supply valve	DR	Vessel rupture disk
P-CO_2	CO_2 pump	V-7	On/off valve
NRV-2	Non-return valve CO_2 line	V-8	Micrometric valve
V-5	CO_2 supply valve to vessel	C	Product collector
V-6	CO_2 supply valve to vessel	-	-

Table 5.1: Parts and components of the 75 mL high pressure plant.

5.4 Instruments, techniques and procedures for the characterization of *Quatsomes*

5.4.1 Cryogenic Transmission Electron Microscopy (Cryo-TEM)

The morphological properties of *Quatsomes* were studied by cryo-TEM. The images were obtained using a *JEOL JEM-2011* tansimission electron microscope (JEOL LTD., Tokyo, Japan), operating at 120 kV. A small drop of the sample was placed on a copper grid coated with a perforated polymer film. Excess solution was thereafter removed by blotting with filter paper, creating a small film of solution. Immediately after film preparation, the grid was plunged into liquid ethane held at a temperature just above its freezing point (94 K). The vitrified sample was then transferred to the microscope for analysis. To prevent sample perturbation and the formation of ice crystals, the specimens were kept at (77 K) during the viewing process.

5.4.2 Dynamic Light Scattering (DLS)

Vesicles size was measured using a dynamic light scattering analyzer combined with non-invasive backscatter technology (Malvern Zetaseizer Nanoseries, Malver Instruments, U.K.). This DLS instrument uses a 4 mW He-Ne laser (λ= 633 nm) and is equipped with a detector angle locked at 173^{o} and a thermostatic sample chamber that allows cooling or heating of the samples. Samples of 1 ml were analyzed without any dilution. The reported size in this master's thesis is the average value of 3 consecutive measurements of the same sample. The particle size distributions have been presented in terms of scattering intensity: they provide information about the % of light scattered by particles, of each size in a sample. DLS measures the time-dependent fluctuations of light scattered from particles experiencing Brownian motion, which results from collisions from suspended particles and solvent molecules. Since the Brownian motion depends on the size of the particles, the rate at which the intensity fluctuations occur also depends on this size. Thus, small particles move faster through the solvent molecules and have faster motions, and this fact causes the intensity to fluctuate more rapidly than in larger particles [29]. Analysis of these intensity fluctuations enables the determination of the diffusion coefficients diffusion coefficient (D) of the particles which are

converted into a size distribution through the Stokes–Einstein equation:

$$d(H) = \frac{kT}{3\pi\eta D} \tag{5.1}$$

where *d(H)* is the hydrodynamic radius of the *Quatsomes* studied, k is the Boltzmann constant, T is the temperature, and η is the solvent viscosity. The hydrodynamic radius of the vesicles is defined as the apparent size solvated sphere.

Regarding the PdI of the samples analyzed, if one were to assume a single size population following a Gaussian distribution, de PdI would be related to the standard deviation, σ, of the hypothetical Gaussian distribution, and the mean size (or Z-average, Z_D). The PdI formula follows:

$$PdI = \frac{\sigma^2}{Z_D^2} \tag{5.2}$$

6 | Conclusions and future works

In this project, the ethanol distribution at the first stages of the depressurization step of DELOS-SUSP process has been studied by CFD for the first time.

On the one hand, *Quatsomes* have been experimentally produced by DELOS-SUSP at different flow rates. Their size, shape and polidispersity have been characterized by DLS and Cryo-TEM techniques. On the other hand, the depressurization step of the DELOS-SUSP has been simulated using OpenFOAM, a free-source CFD tool. In particular, 15 seconds of the process at different flow rates, namely Q = 35 cm^3/s and Q = 175 cm^3/s have been simulated. In order to characterize the mixing dynamics in the simulated system, the concentration of ethanol, in terms of the volume fraction, has been sampled at different times. Several heights and radii have been explored throughout the simulation domain. Moreover, aiming at a comparison of the results at the two different flow rates, the simulated data have been normalized by the total time of the process.

In addition, this work also includes a detailed description of the OpenFOAM simulation setup, which covers: (1) the generation and modification of the geometry and mesh; (2) election of the adequate solver; and (3) setting-up of the boundary conditions and simulation parameters.

Regarding the results, a first approach was to analyze the ethanol volume fraction vs. time (t) for the given period of time (15 s). In this context, a general linear increase of the ethanol concentration for the process at high Q has been observed. As a result, a higher spatial homogeneity in the ethanol distribution can be presumed. This is in good agreement with the agitation produced by the vigorous CO_2 bubbling observed for the simulations at high Q. As expected, the weak agitation related to a lower Q yielded a less homogenous ethanol distribution. This low homogeneity could be related to the formation of ethanol clusters that, in turn, could be the reason of the lower polydispersity observed in the *Quatsomes* experimentally produced at low Q (Figure 4.1).

As previously mentioned, all the collected data from the simulations were taken for t = 15 seconds. Taking into account that two flow rates were considered, a different time to complete the process (tf) was required (i.e. Q = 35 $cm_3/s \rightarrow t_f$ = 600 s, Q = 175 $cm_3/s \rightarrow t_f$ = 120 s). In this sense, a different completed percentage of the process was achieved in each case.

In order to get a direct comparison, a normalization of the results based on t_f was performed (i.e., t / t_f). Consequently, only the 2.5% of the process could be compared with just 15 seconds of simulation time and no general conclusions can be extracted from this data normalization.

As a final conclusion, —considering that: (1) multiphasic modeling of more than two fluids requires extremely long simulation times, even with rader coarse meshes; and (2) OpenFOAM has a steep learning curve— obtaining simulation results in the available task has proven to be a challenging task. Nevertheless, this work lays the groundwork for future studies in this field.

Finally, aiming at a detailed characterization of the process, future work should include: (1) simulations of the full depressurization step; (2) a higher refinement of the mesh in which the simulation domain would be accurately defined; (3) the use of a more suitable solvers that considers the changes in the temperature and turbulent modelization; and (4) the inclusion of the building molecules of the *Quatsomes* (i.e., cholesterol and CTAB) into the simulations in order to accurately describe the system.

Bibliography

[1] Nel A.E., Mädler L., Velegol D., Xia T., Hoek E.M., Somasundaran P., Klaessig F., Castranova V., and Thompson M. **Understanding biophysicochemical interactions at the nano–bio interface**. *Nature Materials*, (8):543–557, 2009.

[2] Robert Langer. **Drug delivery and targeting**. *Nature*, (392):5–10, 1998.

[3] Soussan E., Cassel S, Blanzat M, and Rico-Lattes I. **Drug Delivery by Soft Matter: Matrix and Vesicular Carriers**. *Angew. Chem. Int*, (48):274–288, 2009.

[4] Gregory Gregoriadis, C.P. Swain, E.J. Wills, and A.S. Tavill. **Drug carrier potential of liposomes in cancer chemoterapy.** *The Lancet*, (7870):1313–1316, 1974.

[5] Bangham D. and Horne R.W. **Negative staining of phospholipids and their structural modification by surface-active agents as observed in the electron microscope**. *Journal of Molecular Biology*, (8):660–668, 1964.

[6] Dubois M. and Zemb T. **Phase behaviour and scattering of double-chain surfactants in diluted aqueous solutions**. *Langmuir*, (7): 1352–1360, 1991.

[7] J. Veciana, L. Ferrer-Tasies, E. Moreno-Calvo, M. Cano-Sarabia, M. Aguilella-Arzo, A. Angelova, S. Lesieur, S. Ricart, J. Faraudo, and N. Ventosa. **Quatsomes: Vesicles formed by self-assembly of Sterols and quaternary ammonium surfactants**. *Langmuir*, (22): 6519–6528, 2012.

[8] Ferrer Tasies L. *Cholesterol and Compressed CO2: a Smart Molecular Building Block and Advantageous Solvent to Prepare Stable Self-assembled Colloidal Nanostructures.* Doctoral thesis, Universitat Autònoma de Barcelona. 2016.

[9] Cabrera I. *et al.* **Multifunctional Nanovesicle-Bioactive Conjugates Prepared by a One-Step Scalable Method Using CO2-Expanded Solvents**. *Nano Letters*, (13):3766–3774, July 2013.

[10] Ardizzone A. *et al.* **Nanostructuring Lipophilic Dyes in Water Using Stable Vesicles, Quatsomes, as Scaffolds and Their Use as Probes for Bioimaging**. *Small*, (14):1703851, March 2018.

[11] Jennings J., Beija M., Richez A.P., Cooper S.D., Mignot P.E., Thurecht K.J., Jack K.S., and Howdle S.M. **One-pot synthesis of block copolymers in supercritical carbon dioxide: a simple versatile route to nanostructured microparticles**. *Journal of the American Chemical Society*, (10):4772–4781, 2012.

[12] Pasquali I. and Bettini R. **Are pharmaceutics really going supercritical?** *Int J Pharm*, (2):176–187, 2008.

[13] Veciana J., Sala S., Cordoba A., More-Calvo E., Elizondo E., Muntó M., Rojas P.E., Larrayoz M.A., and Ventosa N. *et al.* **Crystallization of Microparticulate Pure Polymorphs of Active Pharmaceutical Ingredients Using CO2-Expanded Solvents**. *Crystal Growth and Design*, (4):1717–1726, 2012.

[14] Torres J., Ventosa N., Sala S., Veciana J., and Joan Llibre. **Depressurization of an Expanded Liquid Organic Solution (DELOS): A New Procedure for Obtaining Submicron- or Micron-Sized Crystalline Particles**. *Crystal Growth and Design*, (4):299–303, 2001.

[15] Meure L.A., Foster N.R., and Dehghani F. **Conventional and dense gas techniques for the production of liposomes: a review**. *AAPS PharmSciTech*, (3):370–371, 2012.

[16] Veciana J., Cano-Sarabia M., Ventosa N., Sala S., Patiño C., and Arranza R. **Preparation of uniform rich cholesterol unilamellar nanovesicles using CO2- expanded solvents**. *Langmuir*, (6):2433–2437, 2008.

[17] WO0216003 patent licensed to Nanomol Technologies S.L. *Method for precipitating finely divided solid particles.*

[18] WO2006079889 patent licensed to Nanomol Technologies S.L. *Method for obtaining Micro- and Nano-disperse systems.*

[19] Cano Sarabia A.M. *Preparación de materiales moleculares nanoparticulados y dispersos - vesículas y microemulsiones - empleando fluidos comprimidos.* Doctoral thesis, Universitat Autònoma de Barcelona. 2009.

[20] Ozin G. A. abd Hou K., Lotsch B.V., Cademartini L., Puzzo D.P., Scotognella F., Ghadimi A., and Thomson J. *et al.* **Nanofabrication by self-assembly**. *Materials Today*, (5):12–23, 2009.

[21] Sierra-Pallares J., Alonso E., Montequi I., and Cocero M.J. **Particle diameter prediction in supercritical nanoparticle synthesis using threedimensional CFD simulations. Validation for anatase titanium dioxide production**. *Chemical Engineering Science*, (64): 3051 – 3059, April 2009.

[22] Sierra-Pallares J., Marchisio D.L., Parra-Santos M.T., García Serna J., Castro F., and Cocero M.J. **A Computational Fluid Dynamics Study of Supercritical Antisolvent Precipitation: Mixing Effects on Particle Size**. *AIChE Journal*, (58):386–394, February 2012.

[23] Hood R.R., DaVoe D.L., Atencia J., Vreeland W.N., and Omiatek D.M. **A facile route to the synthesis of monodisperse nanoscale liposomes using 3D microfluidic hydrodynamic focusing in a concentric capillary array**. *Lab on a chip*, (14):2365–2586, July 2014.

[24] The OpenFOAM fundation. *OpenFOAM User Guide*. Version 7. 10th July, 2019.

[25] Cfd direct: The architects of openfoam. URL `https://cfd.direct/openfoam/user-guide/v6-mesh/-accessed:02.09.2019`.

[26] Nist: Propiedades termofísicas de sistemas fluidos. URL `https://webbook.nist.gov/chemistry/fluid/-accessed:18.06.2019`.

[27] Jochen Winklemann. *Diffusion in gases, liquids and electrolytes*. Physical Chemistry. 2017.

[28] Dittmar D., Fredenhagen S.B., and Eggers O. **Interfacial tensions of ethanol–carbon dioxide and ethanol–nitrogen. Dependence of the interfacial tension on the fluid density—prerequisites and physical reasoning.** *Chemical Engineering Science*, (5):1223–1233, April 2003.

[29] Pecora R. (Ed.)n. *Dynamic light scattering: applications of photon correlation spectroscopy*. New York; Plenum Press. 1985.

Appendices

A | Troubleshooting

This project was started with no knowledge about OpenFOAM or any other CFD or programming related software. Regarding the time employed in doing this master's thesis, around 60% or 70% of this time was spent either learning how to run the software or troubleshooting the incessant problems and errors that were appearing.

As an intent to ease the path for future works, a brief section covering the most relevant errors and its solutions is presented.

A.1 Flow rate not matching the specified value

Inlet flow is defined in the *Inlet patch* by the velocity boundary condition *fixedValue*, where a specific value for the velocity is defined at the cells belonging to that patch. This statement implies that, unlike other parameters, the velocity in the inlet is **imposed** by the user as a constant value.

However, when testing the system with different mesh refinements, it was realized the inlet flow was insufficient in some of those tests. This fact was firstly noticed by the lack of bubbling at the lower end of the pipe, since in normal conditions, the insufflation of the CO_2 into the aqueous phase generates a very characteristic bubbling. Instead, in these runs, the velocity of the flow in the pipe was barely high enough to make the gas reach the bulk of the aqueous phase.

After some research, it was discovered that a poor refinement of the pipe was causing the abnormalities. The inlet is defined as a bidimensional plane. This plane in turn, is attached to the cylindrical wall of the pipe. As said above, the inlet has a *fixedValue* BC that forces the velocity to a fixed value (either 2.8 m/s or 14 m/s in this work). However, in the wall of the pipe, the *noSlip* condition forces $U = 0$ m/s. These conditions were already defined in Section 3.3. So, as seen in Figure A.1, the cells in the vertex have neither $U = 2.7$ m/s nor $U = 0$ m/s, but a value in-between, as seen in the Figure A.1 by the red-witish colour in the vertices.

When performing preliminar tests, the refinement of the mesh could not be higher since it would have taken a huge processing time. For a multiphasic

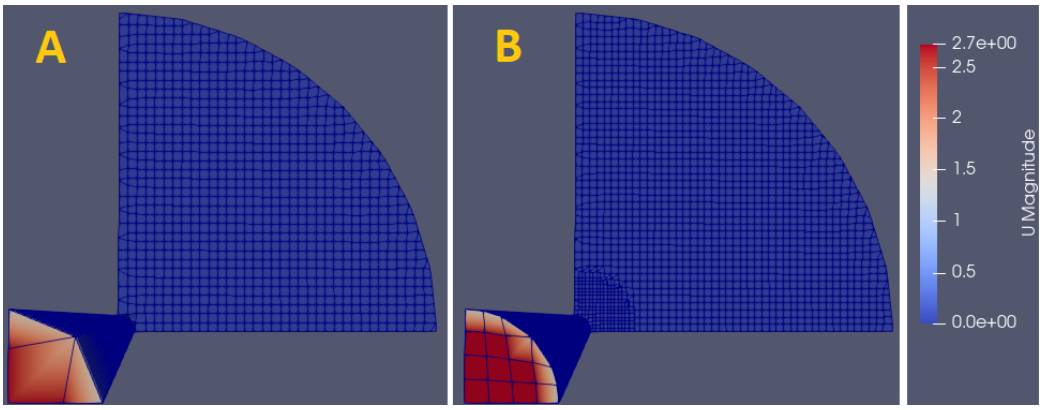

Figure. A.1: Coarse inlet (A) and refined inlet (B). Increasing velocity is shown in color red.

flow, a 5 seconds simulation can take up to a week of processing time for a middle-refined mesh. In the poorly refined meshes, like the one showed in Figure. A.1: *A*, the velocity is not the specified by the user in any cell but lower, yielding a flow with the in-between value mentioned above. This problem was then solved by further refining the mesh, as in Figure A.1: *B*, but at the cost of increasing the simulation time greatly. By refining the mesh, several smaller cells were obtained, some of which were not in contact with the pipe walls, and having, as a result, the desired velocity.

A.2 Destabilization of the system: pressure residuals abnormally high

Another recurrent problem was the destabilization of pressure in the simulation domain. The collector and the pipe worked at a pressure coherent with the boundary conditions, that was calculated by OpenFOAM (normally close to the atmospheric pressure). Sometimes, spontaneously, the pressure sharply increased in the collector and the pipe to a values close to 3 MPa. In turn, in order to maintain constant the velocity in the inlet, the pressure in the inlet would also raise drastically. Eventually, an over pressure in the inlet would be so strong that all the liquid inside the collector would be ejected out of the domain due to the strong "blow" of gas coming from the pipe. The results of this errors can be seen in Figure A.2. In a moment, all the liquid could be lost, making that simulation useless, sometimes after days

processing. This error would happen frequently and at random simulation times.

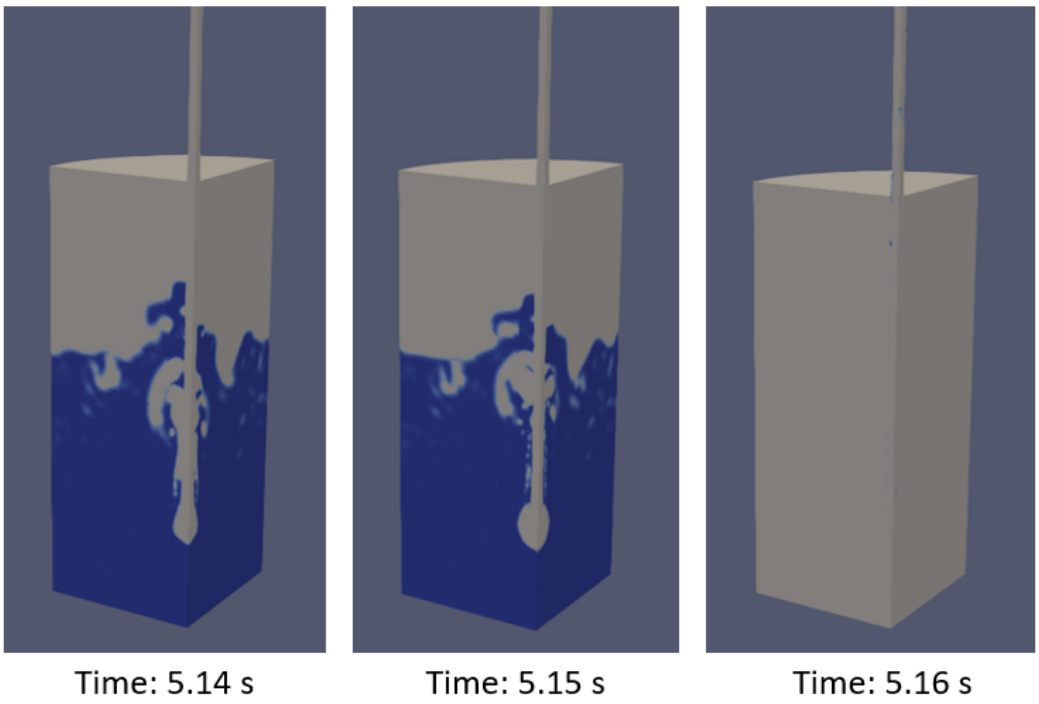

Figure. A.2: Depressurization process over time. In the figure it can be seen that at t 5.16 s, the water has disappeared due to an overpressure in the inlet that causes the aqueous phase to be ejected out of the domain.

The cause of these errors was never fully understood but were blamed on numerical instabilities in the CFD software. Eventually, it was discovered that this problem could be solved by adding extra iterations to the calculus of the pressure equation. By default, OpenFOAM consecutively calculates 3 times the pressure in every cell to avoid any numerical divergences.

In figure A.3: A, the residuals of the **third** iteration of the pressure calculation can be observed. Ideally, this values should be close to 0, but around t = 2.5 s the pressure becomes unstable as can be seen by the sharp increase in the residuals. On the other hand, in Figure A.3: B, the **sixth** iteration of the pressure for the same simulation is shown. After adding 3 extra iterations, the pressure does not destabilize as can be seen in the residuals, that does not experience a sharp increase but only punctual increases that the software can correct.

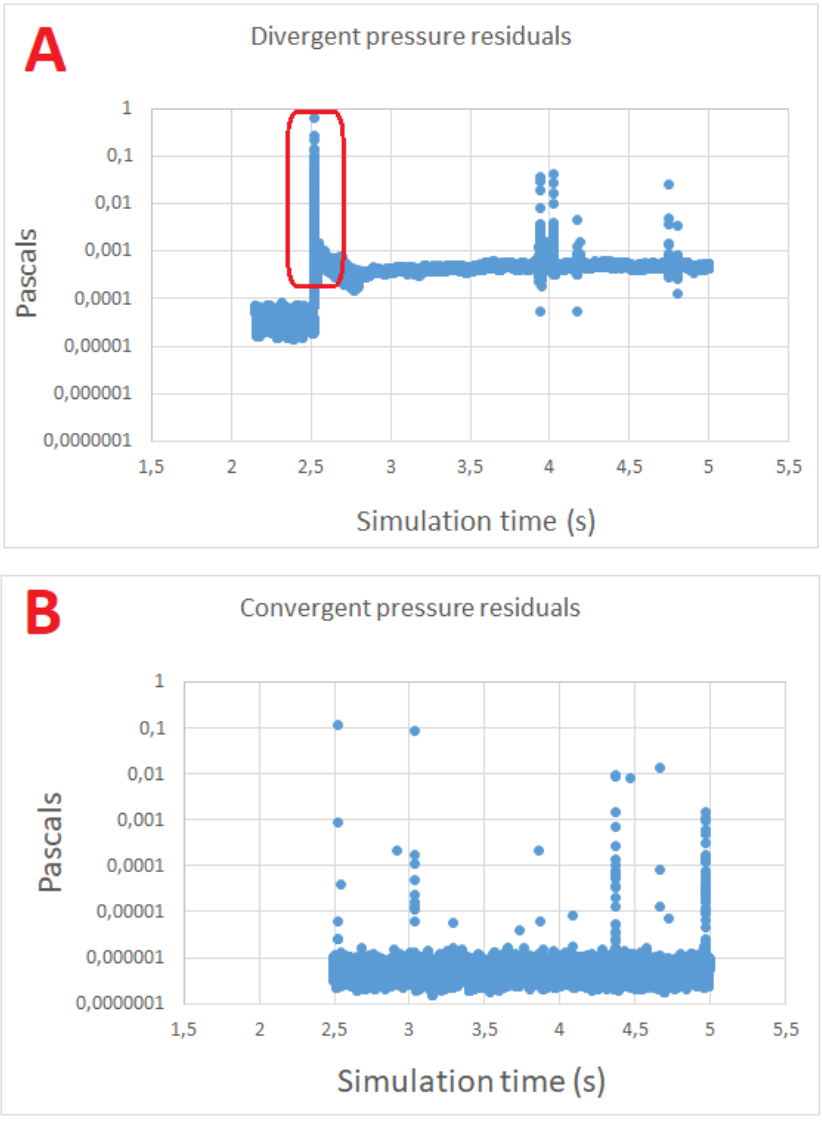

Figure. A.3: Pressure residuals over time for a divergent simulation (A) and a convergent one (B).

B | Dictionaries

In this appendix, the most important dictionaries that define the parameters for the creation of the mesh, and for the simulations are covered. Several other files are needed in order to run those simulations, but it would be impossible to cover a whole case in a single project like the present one. For more detailed information regarding all the files and dictionaries needed for starting up a case in OpenFOAM, one can refer to the web *cfd.direct/openfoam*.

B.1 blockMeshDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.0                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

convertToMeters 1;

xmin -45;
xmax  45;
ymin -45;
ymax  45;
zmin -60;
zmax  216;

convertToMeters 0.001;

vertices
(
    (0  0  $zmin)
    ($xmax  0  $zmin)
    ($xmax  $ymax  $zmin)
    (0  $ymax  $zmin)
    (0  0  $zmax)
    ($xmax  0  $zmax)
    ($xmax  $ymax  $zmax)
    (0  $ymax  $zmax)
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (50 50 300) simpleGrading (1 1 1)//
```


Figure. B.1: *blockMeshDict* structure.

The marked entries in figure B.1 have the following meaning:

1. Scales the length of the properties directly below, since by delfault OpenFOAM works in meters. In the case of this figure, for example, the position of the vertices is scaled to 0.001 m, so their position is given in millimeters.

2. Sets the variables for each minimum and maximum point of the cartesian coordinates.

3. Defines the vertices of the mesh. In this case, there are 8 points so the volume is a hexahedron.

4. The refinement, or in other words, how many cells exist in every direction. The first numbers (0-7) enumerate the vertices that define the block. The *(50 50 300)* parentheses defines the ratio between cells relative to the other directions. (in this case, for every 50 cells in x and y, there are 300 in z). Increasing the refinement will increase the number of cells in our mesh, which in turn will greatly increment the processing time.

B.2 SnappyHexMeshDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.0                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      snappyHexMeshDict;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

castellatedMesh true;
snap            true;
addLayers       false;

geometry
{
    tank.stl
    {
        type triSurfaceMesh;
        name tank;/
    }
    lid.stl
    {
        type triSurfaceMesh;
        name lid;
    }
    pipe.stl
    {
        type triSurfaceMesh;
        name pipe;
    }

    pipeRefined
    {
    type            searchableCylinder;
    point1          (0 0 0.216);
    point2          (0 0 -0.033);
    radius          0.002;
        }
};
```

1

2

Figure. B.2: *SnappyHexMeshDict* structure (part 1).

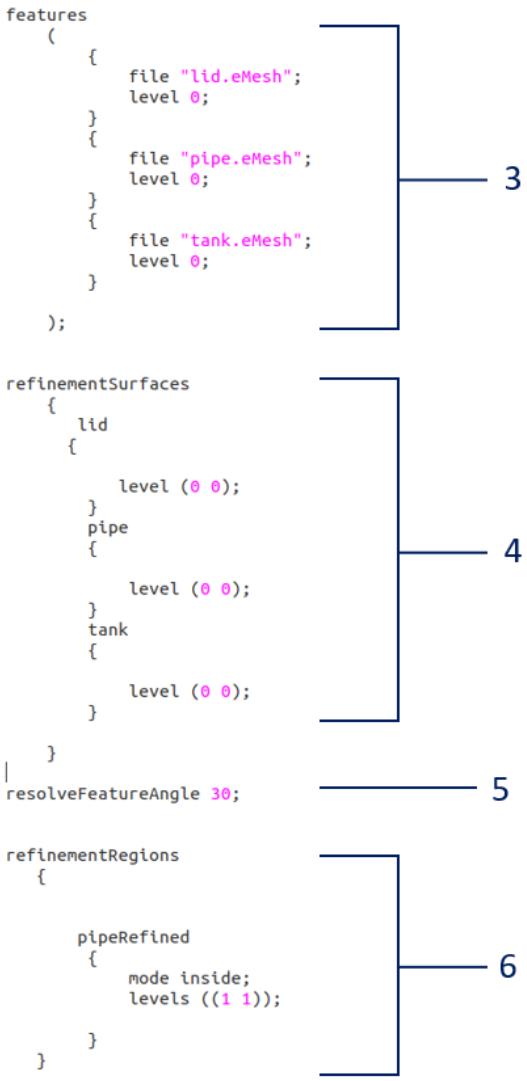

Figure. B.3: *SnappyHexMeshDict* structure (part 2).

The marked entries in figures B.2 and B.3 have the following meaning:

1. In the first step, *castellatedMesh*, the reference base mesh is created (with or without refinements) around the object. In common words, the mesh is created with a "LEGO-like" aspect. Later, in the *snap* step, the castellated mesh is snapped onto the object's surface. Finally, the mesh layers are generated on selected surfaces and adjusted to the main mesh. These three options can be activated or deactivated by *true/false* commands.

2. In here, the files from the *.stl* surfaces are imported to OpenFOAM for snapping the mesh with the desired geometry. A fourth geometry can be added and defined by the user (*pipeRefined*).

3. The *features* field imports *.eMesh* files created with the *surfaceFeature-Extract* command. This files contain the vertices of the *.stl* geometries. These vertices, which contain closed angles, are harder to render, reason why this extra step is needed.

4. Refinement of the mesh where the surfaces are located. Between the parenthesis are two numbers *(0 0)*, zeros in this case. This means that there is no further refinement. Refinement of 1 means that every original cell is divided in 8 cells. Refinement of 2 means that every cell is divided in 64 cells, and so on. The first number of the parenthesis indicates the minimum refinement whereas the second one indicates the maximum.

5. *resolverFeatureAngle* entry is linked with entry number 4. When creating the *.eMesh* files, the software takes as a vertex everything with an angle inferior to 30 degrees.

6. Unlike the *refinementSurfaces* field, in here a refinement of a volume instead of a surface can be done. The region being refined is the one defined by the user in the geometry field (2).

B.3 setFields

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v1812                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      setFieldsDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

defaultFieldValues
(
    volScalarFieldValue alpha.air 1
    volScalarFieldValue alpha.other 0
    volScalarFieldValue alpha.water 0
);

regions
(
    boxToCell
    {
        box (-0.041 -0.041 -0.058) (0.041 0.041 0.0077);
        fieldValues
        (
            volScalarFieldValue alpha.air 0
            volScalarFieldValue alpha.other 0
            volScalarFieldValue alpha.water 1
        );
    }


);

// ************************************************************************* //
```

1

2

Figure. B.4: *SetFields* structure

The marked entries in figures B.4 have the following meaning:

1. In *defaultFieldValues* the initial composition of our whole meshed domain is defined. Specifically, in this case, it is specified that all the domain is initially CO_2 (named as air in the dictionary), as it is marked with a "1".

2. In this field, a box region is defined. It is delimited by the coordinates inside the parenthesis and it is specified that this "box" contains water instead of CO_2.

B.4 Pressure

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.0                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/

dimensions      [1 -1 -2 0 0 0 0];//kg m s K mol A cd

internalField   uniform 0;//initially atmospheric pressure in the entire domain

boundaryField
{
    pipe
    {
        type            fixedFluxPressure;
        value           uniform 0;
    }
    Mirror1
    {
        type            symmetryPlane;

    }

    Mirror2
    {
        type           symmetryPlane;

    }

    Inlet
    {

       type            fixedFluxPressure;
        value           uniform 0;
    }
    lid
    {

        type            totalPressure;
        p0              uniform 0;
    }
    tank
    {
        type            fixedFluxPressure;
     value           uniform 0;
    }

}

// ************************************************************************* //
```

1

2

Figure. B.5: *P* structure.

The marked entries in figures B.5 have the following meaning:

1. The units of the pressure are defined as $\frac{kg \cdot m}{s^{-2}}$.
2. The BCs for the pressure. This topic has already been covered in Section 3.3 of this work.

   There is an dictionary for defining the velocity which is not shown since it is equivalent to the one shown in Figure B.5.

B.5 controlDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v1812                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     interMixingFoam;                    1

startFrom       latestTime;                         2

startTime       0;                                  3

stopAt          endTime;                            4

endTime         5;                                  5

deltaT          0.001;                              6

writeControl    adjustableRunTime;                  7

writeInterval   0.01;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression on;

timeFormat      general;

timePrecision   6;

runTimeModifiable yes;

adjustTimeStep  yes;

maxCo           0.5;
maxAlphaCo      0.5;

maxDeltaT       1;
```

Figure. B.6: *controlDict* structure.

The marked entries in figures B.6 have the following meaning:

1. The solver which is being used is defined.
2. It states when the simulation starts. Even though the initial time is 0, a simulation can be paused and then resumed from "latest time".
3. Indicates the original time of the simulation.
4. Sets when the simulation stops. It is normally at the end time of the simulation but it could be earlier.
5. Defines when the simulation ends.
6. Sets how often OpenFOAM saves the results. In other words, it defines the steps if the simulation.
7. Writes data every *writeInterval* seconds of simulated time, adjusting the time steps to coincide with the *writeInterval* if necessary.

C | Simulation data: Plots

C.1 Simulation results: Ethanol volume fraction *vs.* time.

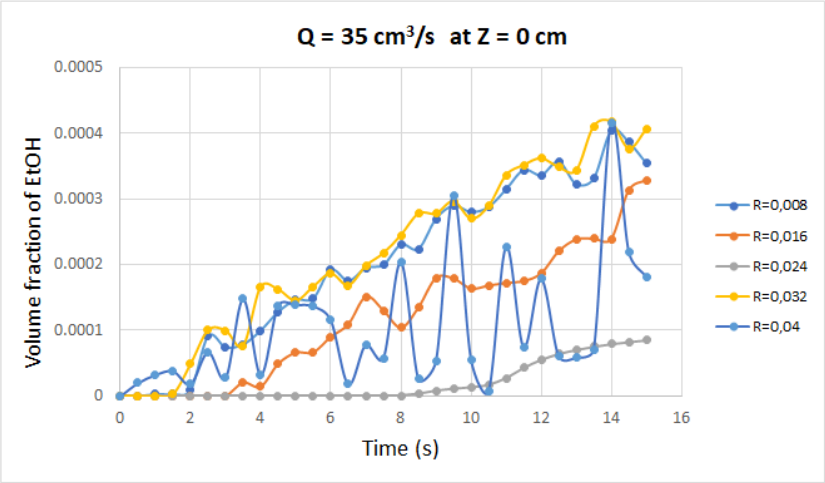

Figure. C.1: EtOH volume fraction *vs.* time for Z = 0 cm and Q = 35 cm^3/s

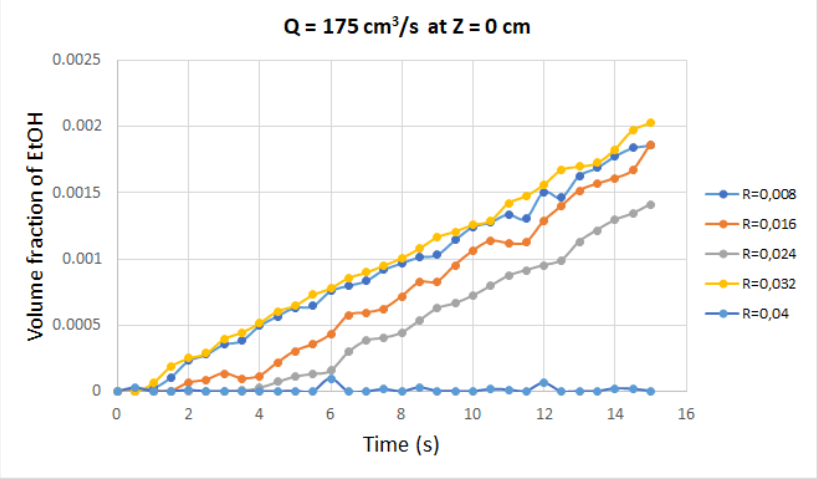

Figure. C.2: EtOH volume fraction *vs.* time for Z = 0 cm and Q = 175 cm^3/s

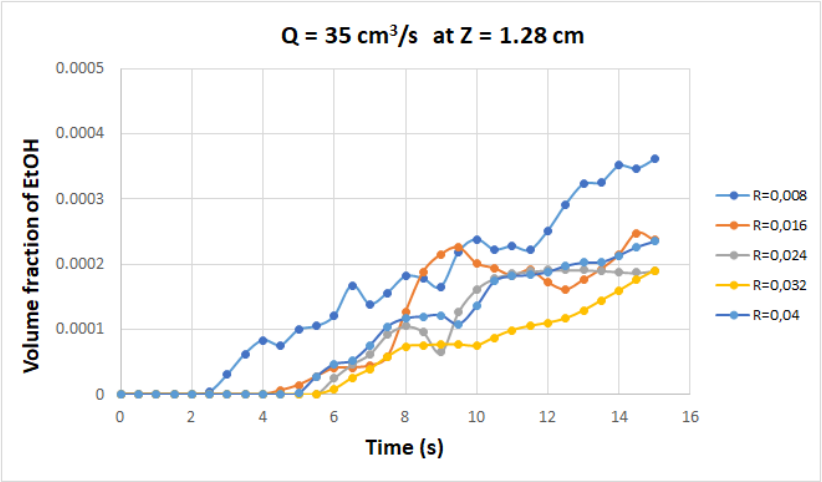

Figure. C.3: EtOH volume fraction *vs.* time for Z = 1.28 cm and Q = 35 cm^3/s

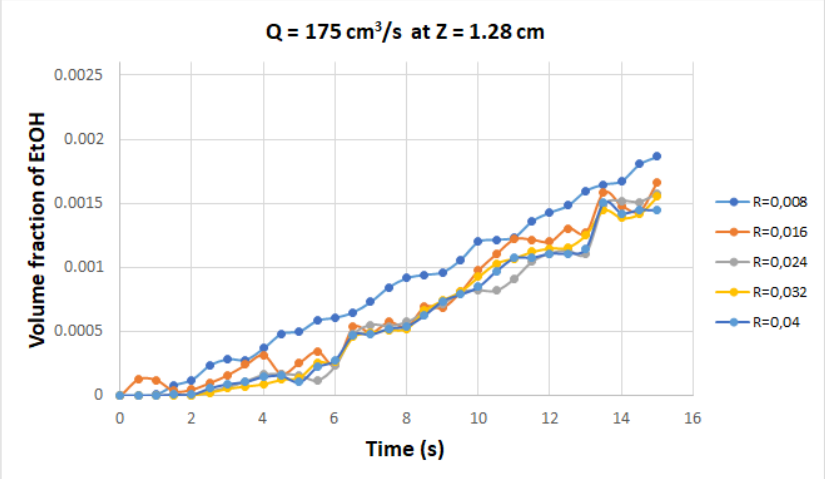

Figure. C.4: EtOH volume fraction *vs.* time for Z = 1.28 cm and Q = 175 cm^3/s

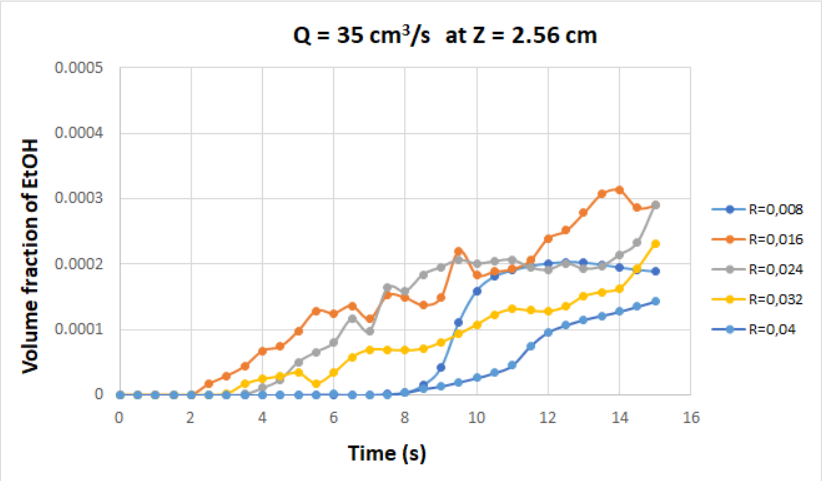

Figure. C.5: EtOH volume fraction *vs.* time for Z = 2.56 cm and Q = 35 cm^3/s

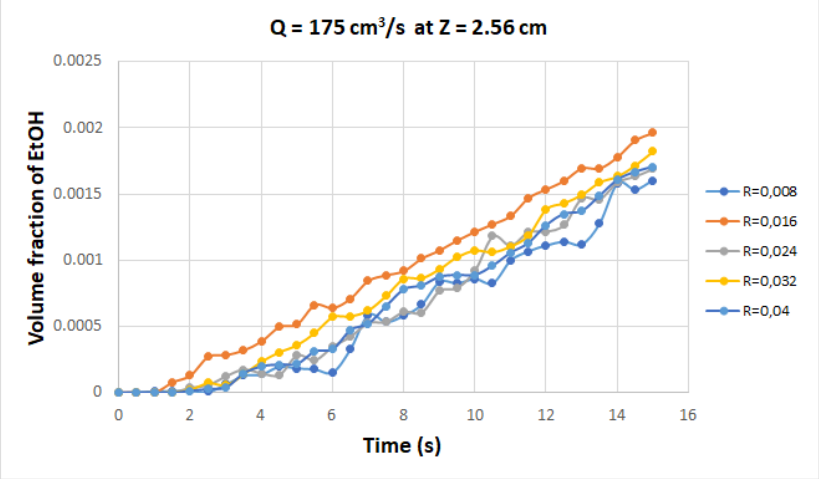

Figure. C.6: EtOH volume fraction *vs.* time for Z = 2.56 cm and Q = 175 cm^3/s

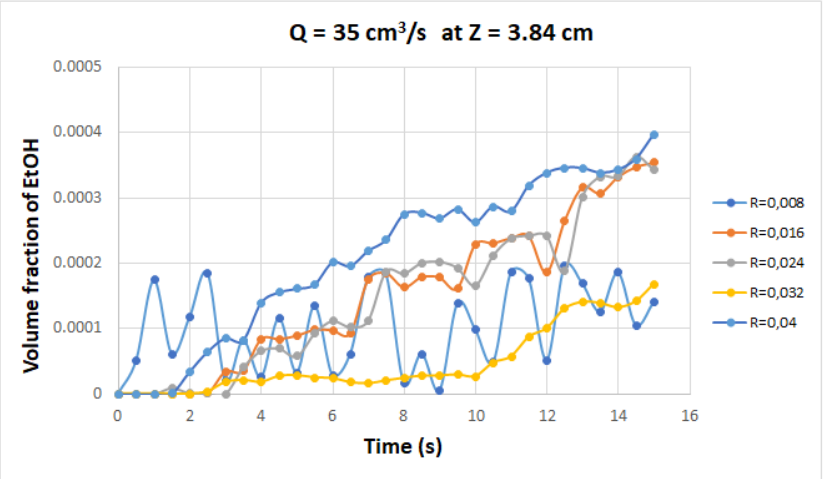

Figure. C.7: EtOH volume fraction *vs.* time for Z = 3.84 cm and Q = 35 cm^3/s

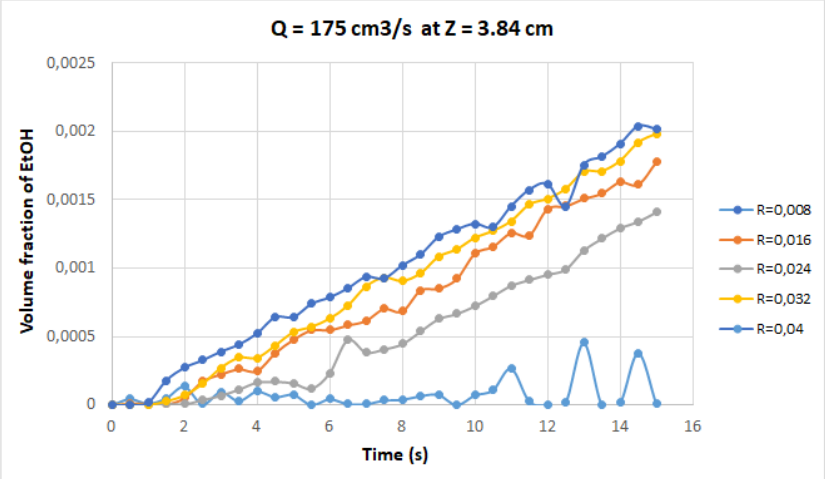

Figure. C.8: EtOH volume fraction *vs.* time for Z = 3.84 cm and Q = 175 cm^3/s

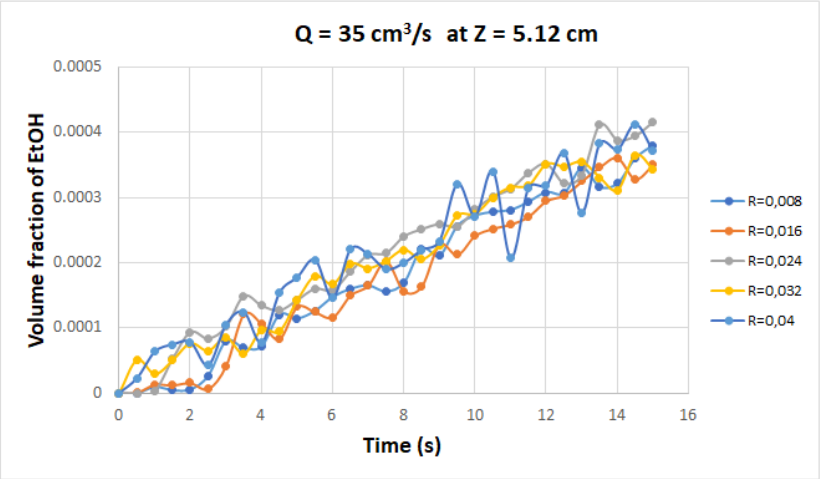

Figure. C.9: EtOH volume fraction *vs.* time for Z = 512 cm and Q = 35 cm^3/s

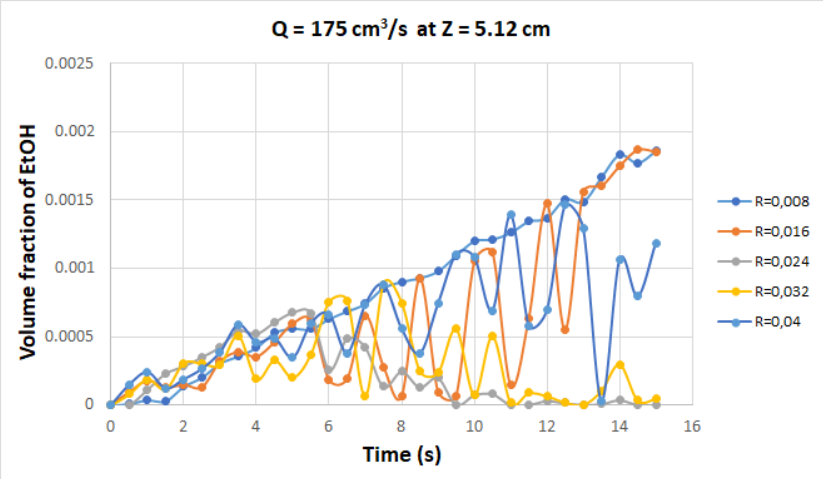

Figure. C.10: EtOH volume fraction *vs.* time for Z = 512 cm and Q = 175 cm^3/s

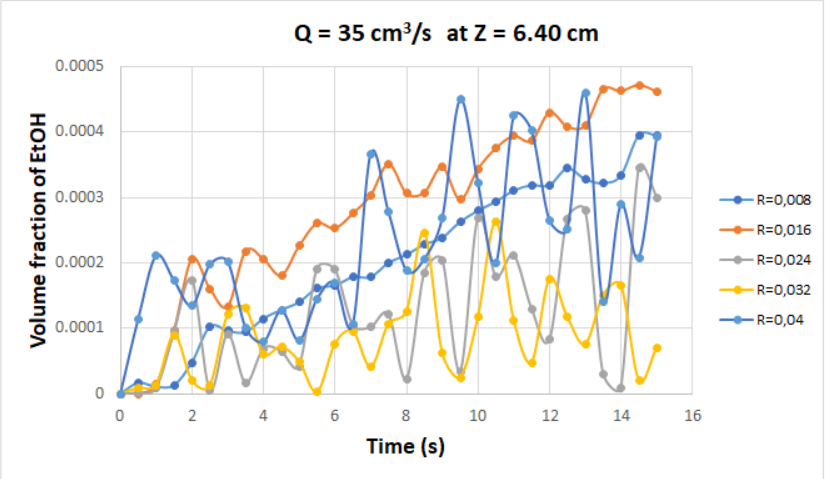

Figure. C.11: EtOH volume fraction *vs.* time for Z = 6.40 cm and Q = 35 cm^3/s

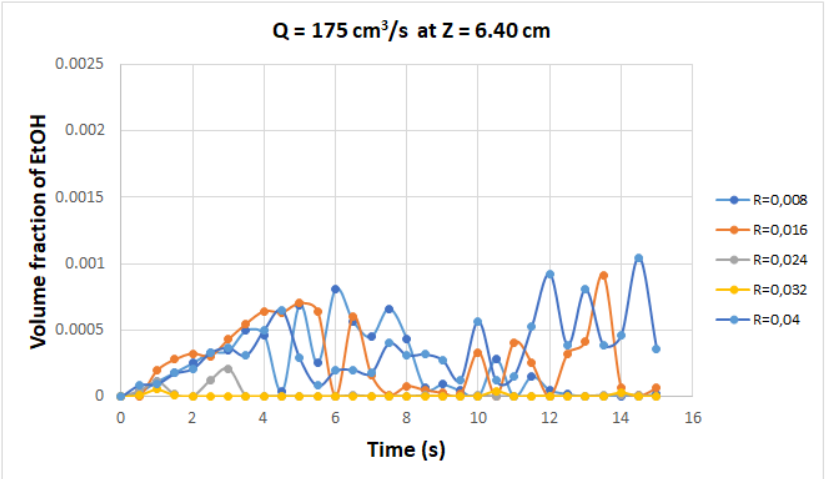

Figure. C.12: EtOH volume fraction *vs.* time for Z = 6.40 cm and Q = 175 cm^3/s